Gestione Fotografie

1 Indice

2 Introduzione

Questo progetto ha come scopo quello di realizzare un applicativo web che permetta di gestire un concorso fotografico.
Ci saranno due tipi di utenti principali, i fotografi e gli utenti normali, qui di seguito sono elencate le funzionalità principali di questi tipi di utenti.

Funzionalità fotografi:

- Caricare le fotografie
- Visualizzare le proprie fotografie caricate
- Visualizzare le statistiche sulle proprie fotografie
- Visualizzare la classifica delle proprie fotografie

Funzionalità utenti normali:

- Inserire, eliminare e modificare i propri commenti
- Inserire, eliminare e modificare le proprie valutazioni

In più sarà possibile accedere all'applicativo anche senza effettuare un login, gli utenti non loggati avranno le seguenti funzionalità:

- Visualizzare catalogo fotografie
- Visualizzare dettagli fotografie
- Ricerca con filtri delle fotografie
- Visualizzare la classifica globale delle fotografie

Tutte le funzionalità degli utenti non loggati saranno disponibili anche per i fotografi e gli utenti normali.

Chi lo desidera potrà andare a creare un account, dopo la registrazione l'utente amministratore del sistema potrà accedere al pannello di controllo e approvare o rifiutare le richieste di creazione degli account.

3 Analisi dei requisiti

Qui di seguito è presente l'analisi dei requisiti che l'applicativo dovrà avere.

Req-001	
Nome	Login
Priorità	1
Versione	1.0
Note	Dovrà essere presente un sistema di login per poter accedere all'applicativo. I visitatori normali potranno visualizzare le fotografie del concorso senza dover effettuare un'iscrizione e di conseguenza un login. Per gli utenti che vogliono lasciare commenti o valutazioni sarà necessario effettuare il login, come pure per i fotografi che desiderano partecipare al concorso caricando i propri scatti.

Req-002	
Nome	Homepage
Priorità	1
Versione	1.0
Note	Nella homepage dovrà essere presentato il concorso e dovrà essere presente una galleria con le fotografie inserite fino a quel momento

Req-003	
Nome	Dettagli fotografia
Priorità	1
Versione	1.0
Note	Cliccando sulla fotografia bisognerà poter visualizzarne i dettagli

Req-004	
Nome	Classifica fotografie
Priorità	1
Versione	1.0
Note	Dovrà essere presente una classifica delle fotografie

Req-005	
Nome	Ricerca fotografie
Priorità	1
Versione	1.0
Note	Dovrà essere possibile effettuare una ricerca delle fotografie in base ai dati utili di quest'ultime (luogo, data, ora, soggetto, b/n o colori, dati fotografo, n° visualizzazioni)

Req-006	
Nome	Iscrizione utente
Priorità	1
Versione	1.0
Note	Un utente che vuole valutare le fotografie dovrà iscriversi immettendo i suoi dati personali

Req-007	
Nome	Conferma iscrizione admin
Priorità	1
Versione	1.0
Note	L'amministratore del sito del concorso fotografico dovrà poter visualizzare le richieste di creazione dell'account, e dovrà poter accettarle o rifiutarle (sia per gli utenti che per i fotografi)

Req-008	
Nome	Inserimento commenti e valutazioni
Priorità	1
Versione	1.0
Note	Un utente dovrà poter inserire dei commenti alle fotografie, e dovrà poter anche inserire delle valutazioni espressa in stelle (da 1 a 5)

Req-009	
Nome	Modifica commenti e valutazioni
Priorità	1
Versione	1.0
Note	Un utente che inserisce un commento o una valutazione dovrà poter avere anche la possibilità di modificarlo/a

Req-010	
Nome	Eliminazione commenti e valutazioni
Priorità	1
Versione	1.0
Note	Un utente che inserisce un commento o una valutazione dovrà poter avere anche la possibilità di eliminarlo/a

Req-011	
Nome	Iscrizione fotografo
Priorità	1
Versione	1.0
Note	Un fotografo dovrà poter iscriversi al sito inserendo i propri dati personali

Req-012	
Nome	Upload delle fotografie
Priorità	1
Versione	1.0
Note	Un fotografo dovrà poter caricare sul sito le sue fotografie fino ad un massimo di 5

Req-013	
Nome	Visualizzazione statistiche fotografie
Priorità	1
Versione	1.0
Note	Un fotografo dovrà poter visualizzare le statistiche sulle sue fotografie riguardanti ad esempio le visualizzazioni, i commenti, le valutazioni, la classifica

4 Progettazione DB

Qui di seguito è presente lo schema entità/relazione del database:

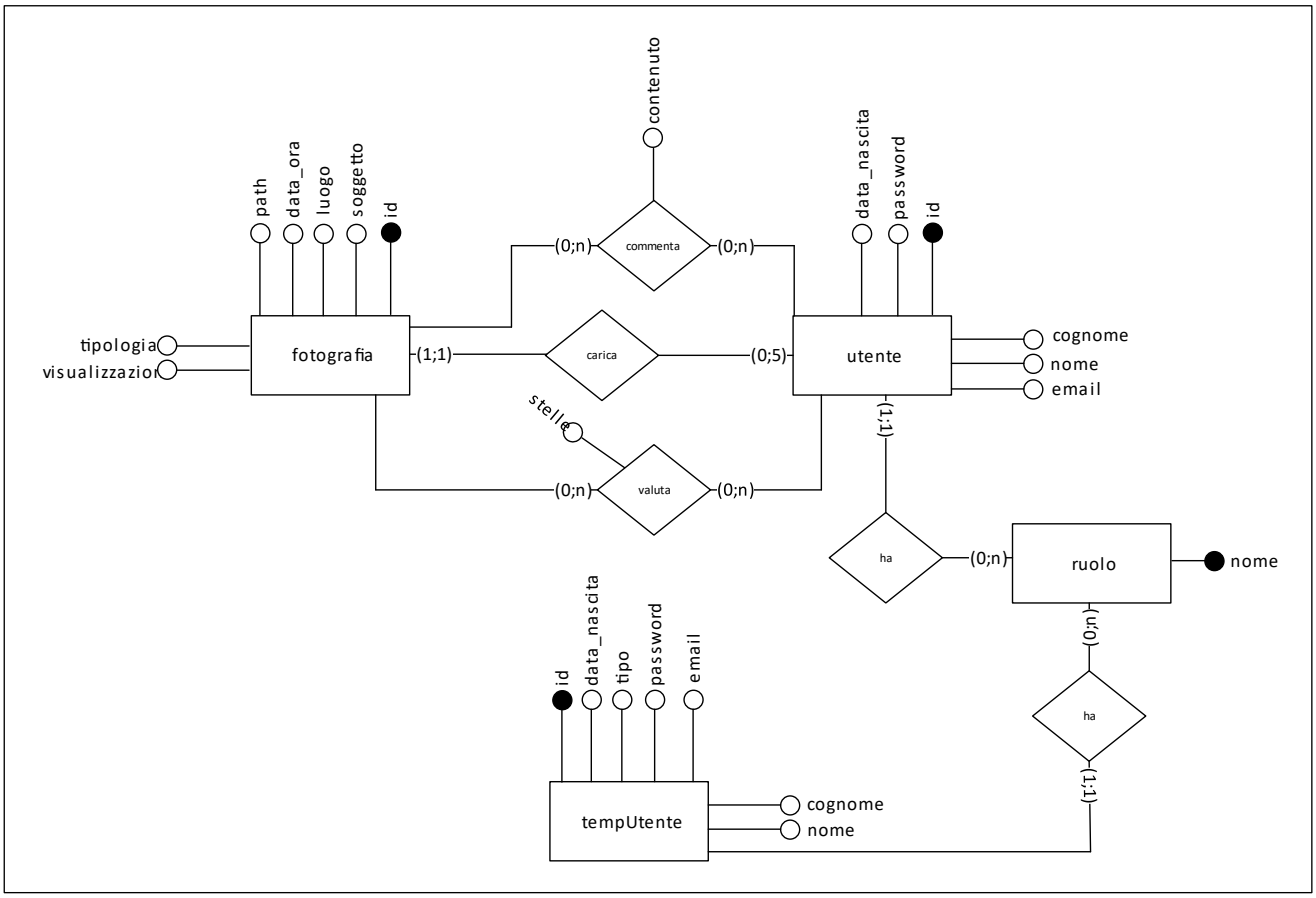

Figura 1 Schema E/R

4.1 Descrizione delle tabelle

4.1.1 Utente

Questa tabella viene utilizzata per memorizzare tutti i dati relativi agli utenti. In questa tabella sono presenti anche le credenziali che vengono utilizzare per effettuare l'accesso all'applicativo (indirizzo email e password).

4.1.2 tempUtente

Questa tabella è strutturata nello stesso modo della tabella utente, e viene utilizzata per memorizzare le informazioni degli utenti che vengono registrati ma di cui l'amministratore non ha ancora approvato la registrazione.
Una volta che l'amministratore approverà la registrazione, i dati verranno spostati nella tabella utente, nel caso invece l'amministratore rifiuti la creazione i dati verranno eliminati.

4.1.3 Ruolo

Questa tabella viene utilizzata per memorizzare il dataset dei ruoli disponibili (amministratore, utente, fotografo).

4.1.4 Fotografia

Questa tabella contiene i dati delle fotografie caricate sull'applicativo. Oltre ai dati relativi ai dettagli della fotografia (data, ora, luogo…) contiene anche il percorso in cui è salvata la fotografia.

4.1.5 Valuta

Questa relazione viene trasformata in una tabella ponte, e si occupa di memorizzare i dati riguardanti le valutazioni degli utenti, come il loro id, l'id della fotografia in cui è stata inserita la valutazione e il numero di stelle.

4.1.6 Commenta

Questa relazione viene trasformata in una tabella ponte, e si occupa di memorizzare i dati relativi ai commenti inseriti dagli utenti. In particolare viene memorizzato l'id del commento, l'id della fotografia a cui è riferito, l'id dell'utente che lo ha creato e il suo contenuto.

5 Implementazione

5.1 Upload delle fotografie – Client

Nella pagina per il caricamento delle fotografie è presente un form con i campi per l'inserimento dei dati riguardanti i dettagli della fotografia, quando viene selezionato un file, viene mostrata una preview dell'immagine selezionata.
Inoltre viene utilizzata una libreria chiamata *piexifjs* che controlla se l'immagine contiene dei dati EXIF, in tal caso verrà estratta la data e l'ora dell'immagine e verrà inserita automaticamente nel campo del form.

```
document.getElementById("formFile").addEventListener("change", async(e) => {
    let file = e.target.files[0];
    if (file) {
        const reader = new FileReader();
        reader.onload = async function (e) {
            previewImage.src = e.target.result;
            document.getElementsByClassName("item-picture")[0].style.display
= "block";

            try{
                // Lettura dati EXIF dell'immagine per ricavare data e ora
                const exifData = piexif.load(previewImage.src);
                const dateTime = exifData["0th"]["306"];
                if(dateTime){
                    dataOraInput.value = formatDateInput(dateTime);
                }
            }catch{}
        };
        reader.readAsDataURL(file);
    }
});
```

Figura 2 Visualizzazione preview immagine durante l'upload

5.2 Upload delle fotografie - Server

Per l'upload delle fotografie è stato creato un controller apposito. Inizialmente prima di processare la richiesta viene controllato se il numero di fotografie caricate dal fotografo non supera il massimo di 5, in seguito vengono fatti dei controlli sul tipo del file caricato (estensione e MIME type) e vengono sanitizzati i dati provenienti dal form.

```
// Gestione dell'upload del file
$fileName = $_FILES["file"]["name"];
Sanitizer::isSetted($fileName);
$fileTmpName = $_FILES["file"]["tmp_name"];
if(!is_uploaded_file($fileTmpName)){
    Twig::render("upload/upload.twig", ["errorMessage" => "Errore durante il
caricamento dell'immagine"]);
    return;
}

$fileRawExt = explode(".", $fileName);
$fileExt = strtolower(end($fileRawExt));
$allowedExt = array("jpg", "jpeg", "png");
if(!in_array($fileExt, $allowedExt)){
    Twig::render("upload/upload.twig", ["errorMessage" => "Formato
dell'immagine .{$fileExt} non valido!"]);
    return;
}

$allowedMime = array("image/jpeg", "image/png");
$mimeType = mime_content_type($fileTmpName);
if(!in_array($mimeType, $allowedMime)){
    Twig::render("upload/upload.twig", ["errorMessage" => "Formato
dell'immagine {$mimeType} non valido!"]);
    return;
}
```

Figura 3 Controlli upload fotografia

In seguito se il file ha superato tutti i controlli, viene generato un nuovo UUID che andrà a comporre il nome del file, successivamente il file verrà spostato nella cartella public/datastore e verranno salvati i dati della fotografia nella tabella dedicata all'interno del database.

```
$uuid = Uuid::uuid4()->toString();
$filePath = "public/datastore/" . $uuid . "." . $fileExt;
move_uploaded_file($fileTmpName, $filePath);

$fotografiaMapper->insert($filePath, $dataOra, $luogo, $soggetto, $tipologia,
0, $_SESSION["utente-id"]);

Twig::render("_templates/successPage.twig", ["successMsg" => "Fotografia
caricata con successo!"]);
```

Figura 4 Salvataggio della fotografia e dei dati nel database

5.3 Ricerca delle fotografie

Per effettuare la ricerca delle fotografie, lato client è presente una maschera in cui si possono selezionare i filtri da utilizzare per la ricerca. Dopo che l'utente ha scritto qualcosa nella barra di ricerca, viene inviata una richiesta tramite AJAX al controller *fotografie*, funzione *search()*.
Nel controller i dati vengono sanitizzati e viene controllato che l'utente non abbia inserito appositamente dei campi non consentiti per la ricerca.

```
$allowedFilters = array("data_ora", "luogo", "soggetto", "tipologia",
"visualizzazioni", "nome_fotografo", "cognome_fotografo");
$sanitizedFilters = array();
foreach($data["filters"] as $filter){
    $filter = Sanitizer::sanitize($filter);
    if(!in_array($filter, $allowedFilters)){//Campo per la ricerca non valido
        $response = array("status" => "FAILED");
        echo json_encode($response);
        return;
    }
    $sanitizedFilters[] = $filter;
}
```

Figura 5 Controllo dei campi per la ricerca

In seguito i dati vengono passati alla funzione *search()* della classe *FotografiaMapper* che si occpua di comporre la query e di ritornare al controller i dati trovati.

```
$selectSql = "SELECT fotografia.*, utente.nome, utente.cognome FROM
fotografia INNER JOIN utente ON fotografia.utente_id=utente.id WHERE ";
$whereSql = array();
for($i = 0; $i < count($filters); $i++){
    if($filters[$i] == "nome_fotografo"){
        $filters[$i] = "utente.nome";
    }
    if($filters[$i] == "cognome_fotografo"){
        $filters[$i] = "utente.cognome";
    }
    if($i == count($filters) - 1){ //Ultimo elemento, non server OR
        $whereSql[] = $filters[$i] . " LIKE :value";
    }else{
        $whereSql[] = $filters[$i] . " LIKE :value OR";
    }
}
$fullQuery = $selectSql . implode(" ", $whereSql);

$stm = $this->db->prepare($fullQuery);
$value = '%' . $value . '%';
$stm->bindParam(":value", $value);
$stm->execute();
$result = $stm->fetchAll();
```

Figura 6 Creazione query per la ricerca delle fotografie - FotografiaMapper

5.4 Inserimento di una valutazione

Per l'inserimento di una valutazione, sono state inserite delle stelle che l'utente può cliccare, in base alla stella cliccata dall'utente, verranno colorate le stelle precedenti.

```
for(let i = 0; i < stars.length; i++){
    stars[i].addEventListener("click", () => {
        resetStars();
        for(let j = 0; j <= i; j++){
            stars[j].src = URL + "public/assets/images/star_filled.png";
            starsCount++;
        }
    });
}
```

Figura 7 Colorazione delle stelle al click dell'utente

In seguito quando l'utente clicca sul pulsante per salvare la valutazione, viene inviata una richiesta al server tramite AJAX con un JSON che contiene il quantitativo di stelle inserito.

```
addValutazione.addEventListener("click", async() => {
    if(starsCount > 0){
        const data = {
            "foto_id":fotoId,
            "stelle":starsCount,
            "action":"insert",
            "CSRFToken":CSRFToken
        };
        let response = await fetch(URL + "fotografie/valuta", {
            method: "POST",
            headers: {
                "Content-Type":"application/json"
            },
            body: JSON.stringify(data)
        });
        let json = await response.json();
        if(json.status == "SUCCESS"){
            Swal.fire({
                title: "Valutazione aggiunta con successo!",
                icon: "success"
            }).then(() => {
                window.location.reload();
            });
        }else{
            Swal.fire({
                title: "Errore nell'inserimento della valutazione",
                icon: "error"
            });
        }
    }else{
        Swal.fire({
            title: "Impossibile aggiungere valutazione con 0 stelle!",
            icon: "error"
        });
    }
});
```

Figura 8 Richiesta AJAX inserimento valutazione

Una volta che la richiesta giunge alla funzione *valuta()* del controller *fotografie* vengono fatti i controlli sul numero di stelle inserito, e che la valutazione non sia dello stesso fotografo a cui appartiene la fotografia.

5.5 Sistema conteggio visualizzazioni fotografie

Per contare il numero di visualizzazioni di una fotografia, è stato creato un sistema che permette ad un utente di visualizzare una sola volta la fotografia per sessione.
All'avvio della sessione viene creata una variabile che ospiterà tutti gli id delle fotografie visualizzate nel corso della sessione.

```php
if (!isset($_SESSION["photos-seen"])) {
    /**
     * Array per la memorizzazione delle fotografie visualizzate
     * la visualizzazione dell'utente viene conteggiata unoicamente una
     * volta per sessione
     */
    $_SESSION["photos-seen"] = array();
}
```

Figura 9 Inizializzazione variabile di sessione per tracciare le fotografie visualizzate

In seguito ogni volta che viene aperta la pagina dei dettagli di una fotografia, viene controllata la variabile di sessione, e se l'id della fotografia non è contenuto in quest'ultima viene incrementato il conteggio delle visualizzazioni.

```php
$photosSeen = $_SESSION["photos-seen"];
if(!in_array($id, $photosSeen)){
    $photosSeen[] = $id;
    $_SESSION["photos-seen"] = $photosSeen;
    $fotografiaMapper->incrementViews($id);
}
```

Figura 10 Controllo incremento delle visualizzazioni

5.6 Sicurezza

Per garantire la protezione da attacchi CSRF, all'inizio della sessione viene generato un token che successivamente viene controllato per ogni richiesta, indipendentemente se essa proviene da un form o da una richiesta AJAX.

```php
public static function genCSRFtoken(){
    $token = bin2hex(random_bytes(35));
    $_SESSION["CSRFToken"] = $token;
}
```

Figura 11 Funzione genCSRFToken - classe Session

Nella classe *Session* è presente anche una funzione per validare il token. In questa funzione bisogna specificare se il CSRF token arriva da una richiesta AJAX, perché in tal caso non potrà essere letto dalla variabile *$_POST* ma dovrà essere passato alla funzione manualmente.

```php
public static function validateCSRFToken($fromJson = false, $jsonCSRFToken =
null){
    if($fromJson){
        if(!isset($_SESSION["CSRFToken"])){
            return false;
        }
        if(!isset($jsonCSRFToken)){
            return false;
        }
        if ($jsonCSRFToken != $_SESSION["CSRFToken"]) {
            return false;
        }
        return true;
    }else{
        if(!isset($_SESSION["CSRFToken"])){
            return false;
        }
        if(!isset($_POST["CSRFToken"])){
            return false;
        }
        if ($_POST["CSRFToken"] != $_SESSION["CSRFToken"]) {
            return false;
        }
        return true;
    }
}
```

Figura 12 Funzione validateCSRFToken - classe Session

Per prevenire il rischio di SQL Injections ogni valore che arriva all'utente viene sanitizzato tramite la classe *Sanitizer*, in aggiunta per effettuare le query vengono utilizzati i prepared statements con il parameter binding.

```
public function insertValutazione($fotografia_id, $utente_id, $stelle){
    $stm = $this->db->prepare("INSERT INTO valuta(fotografia_id, utente_id,
stelle) VALUES(:fotografia_id, :utente_id, :stelle)");
    $stm->bindParam(":fotografia_id", $fotografia_id);
    $stm->bindParam(":utente_id", $utente_id);
    $stm->bindParam(":stelle", $stelle);
    $stm->execute();
}
```

Figura 13 Esempio query con prepared statements

Anche per prevenire il rischio di XSS tutti i dati che arrivano dall'utente vengono sanitizzati, in aggiunta la sessione è stata configurata in modo che scada alla chiusura del browser, in più il cookie non è accessibile da codice JS lato client e viene inviato solo nelle richieste che provengono dallo stesso sito.

```
session_set_cookie_params([
    'lifetime' => 0,
    'path' => '/',
    'httponly' => true,
    'samesite' => 'Strict',
]);
```

Figura 14 Configurazione cookie della sessione

5.7 Unit test

Tutte le funzione di tutte le classi che si occupano di interagire con il database (leggere, scrivere o eliminare dati), sono state testate con PHPUnit.
Per non compromettere il DB dell'applicativo è stato creato uno script *db_test.sql* con già dei dati da utilizzare per i test. Inoltre prima di ogni test viene fatta partire una transazione e alla fine del test viene fatto un rollback, in modo da non compromettere i dati per i test successivi.

```php
public static function setUpBeforeClass(): void
{
    // Caricare variabili per connessione DB da tests/.env
    $dotenv = Dotenv\Dotenv::createImmutable(__DIR__ . '/');
    $dotenv->load();
}

protected function setUp(): void
{
    $this->fotografiaMapper = new FotografiaMapper();
    $this->db = Database::getConnection();
    $this->db->query("START TRANSACTION");
}

protected function tearDown(): void
{
    $this->db->query("ROLLBACK");
}
```

Figura 15 Configurazioni per Unit test

```
davide@davide-pc:/var/www/html/mount/M151/GestioneFotografie$ ./vendor/bin/phpunit tests/
PHPUnit 11.1.3 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.2.18

.................................................               51 / 51 (100%)

Time: 00:00.024, Memory: 8.00 MB

OK (51 tests, 109 assertions)
```

Figura 16 Risultati degli Unit test

6 Test

6.1 Protocollo di test

Test Case **Riferimento**	TC-001 REQ-001	**Nome**	Login
Descrizione	Verificare che si possa effettuare un login come utente, fotografo o amministratore		
Procedura	1. Aprire la homepage dell'applicativo 2. Cliccare sulla sezione "Login/Register" nella barra di navigazione 3. Effettuare il login con un account utente, fotografo e amministratore		
Risultati attesi	Verificare il buon funzionamento del login		

Test Case **Riferimento**	TC-002 REQ-002	**Nome**	Homepage
Descrizione	Verificare che nella homepage sia presente la galleria con le foto del concorso		
Procedura	1. Aprire la homepage dell'applicativo		
Risultati attesi	Nella homepage deve esserci la galleria con le fotografie del concorso		

Test Case **Riferimento**	TC-003 REQ-003	**Nome**	Dettagli fotografia
Descrizione	Verificare che si possano vedere i dettagli di una fotografia		
Procedura	1. Aprire la homepage dell'applicativo 2. Cliccare su una fotografia presente nella galleria		
Risultati attesi	Dovrà apparire una maschera con tutte le informazioni relative alla fotografia cliccata		

Test Case **Riferimento**	TC-004 REQ-004	**Nome**	Classifica fotografie
Descrizione	Verificare che sia presente e che si possa accedere alla classifica delle fotografie		
Procedura	1. Aprire la homepage dell'applicativo 2. Cliccare sulla sezione "Classifica" nella barra di navigazione		
Risultati attesi	Si dovrà poter accedere ad una pagina contenente la classifica delle fotografie		

Test Case **Riferimento**	TC-005 REQ-005	**Nome**	Ricerca fotografie
Descrizione	Verificare che sia possibile cercare le fotografie in base al luogo, data, ora, soggetto, b/n o colori, dati fotografo, n° visualizzazioni		
Procedura	1. Aprire la homepage dell'applicativo 2. Cliccare sulla sezione "Catalogo fotografie" nella barra di navigazione 3. Effettuare la ricerca per ogni tipo di dato descritto sopra		
Risultati attesi	La ricerca deve essere funzionante per ogni tipo di dato		

Test Case **Riferimento**	TC-006 REQ-006	**Nome**	Iscrizione utente
Descrizione	Verificare che un utente possa registrarsi immettendo i suoi dati personali		
Procedura	1. Aprire la homepage dell'applicativo 2. Cliccare sulla sezione "Login/Register" nella barra di navigazione 3. Inserire i dati dell'utente nel form di registrazione 4. Inviare il form		
Risultati attesi	Dovrà essere visualizzato un messaggio di successo che conferma la buona riuscita della registrazione		

Test Case **Riferimento**	TC-007 REQ-007	**Nome**	Conferma iscrizione admin
Descrizione	Verificare che l'amministratore possa visualizzare la pagina con tutte le richieste di creazione degli utenti, e che possa accettare o rifiutare quest'ultime		
Procedura	1. Aprire la homepage dell'applicativo 2. Cliccare sulla sezione "Pannello di controllo" nella barra di navigazione 3. Accettare la richiesta di creazione di un utente 4. Rifiutare la richiesta di creazione di un altro utente		
Risultati attesi	Si dovrà poter accedere alla pagina con le richieste di creazione degli utenti, l'utente a cui è stata accettata la richiesta dovrà poter accedere al sito, mentre l'utente a cui la richiesta è stata rifiutata non dovrà poter accedere		

Test Case **Riferimento**	TC-008 REQ-008	**Nome**	Inserimento commenti e valutazioni
Descrizione	Verificare che l'utente possa inserire delle valutazioni e dei commenti alle varie fotografie		
Procedura	1. Aprire la homepage dell'applicativo 2. Selezionare una fotografia e cliccare su "Visualizza dettagli" 3. Inserire una valutazione 4. Inserire un commento		
Risultati attesi	Si dovrà poter inserire un commento e una valutazione, che dovranno rimanere salvati		

Test Case Riferimento	TC-009 REQ-009	**Nome**	Modifica commenti e valutazioni
Descrizione	Verificare che l'utente possa modificare eventuali commenti o valutazioni da lui inserite		
Procedura	1. Aprire la homepage dell'applicativo 2. Selezionare una fotografia e cliccare su "Visualizza dettagli" 3. Modificare la valutazione 4. Modificare il commento		
Risultati attesi	La valutazione e il commento dovranno essere stati modificati		

Test Case Riferimento	TC-010 REQ-010	**Nome**	Eliminazione commenti e valutazioni
Descrizione	Verificare che l'utente possa eliminare eventuali commenti o valutazioni da lui inserite		
Procedura	5. Aprire la homepage dell'applicativo 6. Selezionare una fotografia e cliccare su "Visualizza dettagli" 7. Eliminare la valutazione 8. Eliminare il commento		
Risultati attesi	La valutazione e il commento dovranno essere stati eliminati		

Test Case Riferimento	TC-011 REQ-011	**Nome**	Iscrizione fotografo
Descrizione	Verificare che un fotografo possa iscriversi		
Procedura	1. Aprire la homepage dell'applicativo 2. Cliccare sulla sezione "Login/Register" nella barra di navigazione 3. Inserire i dati del fotografo nel form di registrazione 4. Inviare il form		
Risultati attesi	Dovrà essere visualizzato un messaggio di successo che conferma la buona riuscita della registrazione		

Test Case **Riferimento**	TC-012 REQ-012	**Nome**	Upload delle fotografie
Descrizione	Verificare che un fotografo possa caricare sull'applicativo fino ad un massimo di 5 fotografie		
Procedura	1. Aprire la homepage dell'applicativo 2. Cliccare sulla sezione "Carica foto" nella barra di navigazione 3. Inserire il file della fotografia che si vuole caricare 4. Inviare il form 5. Caricare altre fotografie fino a raggiungere il limite		
Risultati attesi	Dovrà essere visualizzato un messaggio di successo che conferma la riuscita del caricamento della fotografia, quando si arriva a 5 fotografie non dovrà più essere possibile caricarne di nuove		

Test Case **Riferimento**	TC-013 REQ-013	**Nome**	Visualizzazione statistiche fotografie
Descrizione	Verificare che un fotografo possa visualizzare le statistiche delle sue fotografie caricate		
Procedura	1. Aprire la homepage dell'applicativo 2. Cliccare sulla sezione "Profilo" nella barra di navigazione 3. Andare nella sezione "Statistiche"		
Risultati attesi	Verificare che vengano visualizzate le statistiche per numero di visualizzazioni e per valutazioni		

6.2 Risultati test

Per visualizzare la comprova dei test vedere il video allegato in cui viene mostrato lo svolgimento dei test case.

Test Case	Risultato ottenuto	Stato
TC-001	Tutti gli utenti (amministratore, fotografo, utente) riescono ad effettuare il login	Passato
TC-002	Viene visualizzata la homepage con il catalogo delle fotografie caricate fino a quel momento	Passato
TC-003	Vengono visualizzati i dettagli della fotografia cliccata	Passato
TC-004	Viene mostrata la classifica, sia per le fotografie più votate che per numero di visualizzazioni	Passato
TC-005	La ricerca delle fotografie funziona per ogni campo (data, ora, luogo, tipologia…)	Passato
TC-006	L'utente riesce ad iscriversi e viene visualizzato un messaggio di successo	Passato
TC-007	L'amministratore riesce a visualizzare le richieste di registrazione nel pannello di controllo, rifiutando una richiesta e approvandone un'altra, l'utente rifiutato non riesce ad effettuare il login, mentre quello approvato riesce ad accedere senza problemi	Passato
TC-008	L'utente riesce ad inserire sia i commenti che le valutazioni, e vengono salvati	Passato
TC-009	L'utente riesce a modificare i suoi commenti e valutazioni	Passato
TC-010	L'utente riesce ad eliminare i suoi commenti e valutazioni	Passato
TC-011	Il fotografo riesce ad iscriversi e viene visualizzato un messaggio di successo	Passato
TC-012	Il fotografo riesce a caricare le sue fotografie, fino ad un massimo di 5, oltre quel numero il caricamento non è più consentito	Passato
TC-013	Il fotografo riesce a visualizzare le statistiche delle sue fotografie, sia per numero di visualizzazioni che per il punteggio delle valutazioni	Passato

7 Conclusioni

In conclusione tutte le funzionalità per il funzionamento di base sono state implementate, e tutti i requisiti sono stati soddisfatti.

Sicuramente l'applicativo può essere migliorato a livello estetico e anche per quanto riguarda l'aspetto del responsive.

Penso che in futuro si possano sicuramente aggiungere delle funzionalità in più che rendono l'utilizzo dell'applicativo migliore, come ad esempio:

- Sistema email per conferma registrazione e abilitazione account
- Possibilità di modificare le informazioni del proprio profilo
- Possibilità per i fotografi di eliminare le fotografie caricate
- Inserimento del timestamp per i commenti
- Possibilità per l'amministratore di avere una panoramica completa di tutti gli utenti registrati
- Possibilità per l'amministratore di eliminare commenti con contenuti offensivi o anche immagini offensive

8 Indice delle figure