

# Indice :

1 Requisiti 3

1.1 Specifica generale 3

1.2 Analisi specifica generale 3

1.3 Estensione JSP 3

1.4 Analisi specifiche JSP 3

1.5 Scelte progettuali 3

1.5 Scelte progettuali

1.5.1 Versione generale 6

1.5.2 Versione RIA 6

2 Progettazione base di dati 7

2.1 Schema ER 7

2.2 Struttura del database 7

2.2 Struttura del database

2.2.1 User table 7

2.2.2 Album table 7

2.2.3 Image table 7

2.2.3 ImageAlbumLink table 7

2.2.4 Comment table 7

3 Versione HTML pura 9

3.1 Rappresentazione IFML 9

3.2 Struttura codice 9

3.3 Sequence diagrams 9

3.2 Struttura codice

3.2.1 Model objects (beans) 9

3.2.2 Data Access Objects (DAO) 9

3.2.3 Controller (Servlets) 9

3.2.4 Filters 9

3.2.4 Views 9

3.3 Sequence diagrams

3.3.1 Gestione cambio pagine 11

3.3.2 Gestione eventi nelle singole pagine 11

3.3.1 Gestione cambio pagine

3.3.1.1 Comportamento login 11

3.3.1.2 Comportamento registrazione nuovo utente 11

3.3.1.3 Selezione di un album dalla Home Page 11

3.3.1.4 Selezione di una immagine dall’Album Page 11

3.3.2 Gestione eventi nelle singole pagine

3.3.2.1 Eventi disponibili dalla Home Page 14

3.3.2.2 Eventi disponibili dall’Album Page 14

3.3.2.3 Eventi disponibili dalla Image Page 14

4 Versione RIA 19

4.1 Rappresentazione IFML 19

4.2 Struttura codice 19

4.3 Sequence diagrams 19

4.2 Struttura codice

4.2.1 Model objects (beans) 20

4.2.2 Data Access Objects (DAO) 20

4.2.3 Controller (Servlets) 20

4.2.4 Filters 20

4.2.5 Views 20

4.3 Sequence diagrams

4.3.1 Gestione cambio schermate 22

4.3.1.1 Comportamento login 22

4.3.1.2 Comportamento registrazione nuovo utente 22

4.3.1.3 Selezione di un album dalla Home Page 22

4.3.1.4 “Mouseover” su un’immagine dall’Album Page 22

4.3.1.5 “Mouseleave” dalla finestra modale Image Page 22

4.3.2 Gestione eventi

4.3.2.1 Eventi disponibili dall’Album Page 26

4.3.2.2 Eventi disponibili dalla Home Page 26

4.3.2.3 Eventi disponibili dall’Image Page 26

# 1 Requisiti

## 1.1 Specifica generale

Delle tre tipologie di progetto a me è stato assegnato l’esercizio numero1: “Gestore di immagini”. La specifica per la versione base (HTML puro) era la seguente:

“Un’applicazione web consente la gestione di una galleria d’immagini. L’applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione controlla la validità sintattica dell’indirizzo di email e l’uguaglianza tra i campi “password” e “ripeti password”. La registrazione controlla l’unicità dello username. Ogni immagine è memorizzata come file nel file system del server su cui l’applicazione è rilasciata. Inoltre nella base di dati sono memorizzati i seguenti attributi: un titolo, una data di creazione, un testo descrittivo e il percorso del file dell’immagine nel file system del server. Le immagini sono associate all’utente che le carica. L’utente può creare album dalla HOME PAGE e associare a questi le proprie immagini. Un album ha un titolo, il creatore e la data di creazione. La stessa immagine può appartenere a più di un album. Le immagini sono associate a uno o più commenti inseriti dagli utenti (dal proprietario o da altri utenti). Un commento ha un testo e il nome dell’utente che lo ha creato. Quando l’utente accede all’HOME PAGE, questa presenta l’elenco degli album che ha creato e l’elenco degli album creati da altri utenti. Entrambi gli elenchi sono ordinati per data di creazione decrescente. Quando l’utente clicca su un album che appare negli elenchi della HOME PAGE, appare la pagina ALBUM PAGE che contiene inizialmente una tabella di una riga e cinque colonne. Ogni cella contiene una miniatura (thumbnail) e il titolo dell’immagine. Se il numero di immagini non è un multiplo di 5 la tabella deve avere sempre 5 celle, lasciando quelle più a destra vuote. Le miniature sono ordinate da sinistra a destra per data decrescente. Se l’album contiene più di cinque immagini, sono disponibili comandi per vedere il precedente e successivo insieme di cinque immagini. Se la pagina ALBUM PAGE mostra il primo blocco d’immagini e ne esistono altre successive nell’ordinamento, compare a destra della riga il bottone SUCCESSIVE, che permette di vedere le successive cinque immagini. Se la pagina ALBUM PAGE mostra l’ultimo blocco d’immagini e ne esistono altre precedenti nell’ordinamento, compare a sinistra della riga il bottone PRECEDENTI, che permette di vedere le cinque immagini precedenti. Se la pagina ALBUM PAGE mostra un blocco d’immagini e ne esistono altre precedenti e successive nell’ordinamento, compare a destra della riga il bottone SUCCESSIVE, che permette di vedere le successive cinque immagini, e a sinistra il bottone PRECEDENTI, che permette di vedere le cinque immagini precedenti. Quando l’utente seleziona una miniatura, una pagina IMAGE PAGE mostra tutti i dati dell’immagine scelta, tra cui la stessa immagine a grandezza naturale e i commenti eventualmente presenti. La pagina mostra anche una form per aggiungere un commento e un bottone per cancellare l’immagine e tutti i commenti ad essa associati. Il bottone di cancellazione appare solo se l’immagine appartiene all’utente. L’invio del commento con un bottone INVIA ripresenta la pagina IMAGE PAGE, con tutti i dati aggiornati della stessa immagine. La pagina IMAGE PAGE contiene collegamenti per tornare all’HOME PAGE e alla pagina ALBUM PAGE. La pagina ALBUM PAGE contiene un collegamento per tornare all’HOME PAGE. L’applicazione consente il logout dell’utente.”

## 1.2 Analisi specifica generale

La mia interpretazione della specifica sopra riportata mi ha condotto a immaginare una struttura base composta da quattro elementi che interagiscono tra loro: utente, immagine, album, commento. In particolare:

1. l’utente disporrà dell’attributo email (che per assunzione ho deciso di rendere unico), password e un username unico. A questi ho deciso di aggiungere un id univoco per l’identificazione, sebbene fosse già presente un campo unico (username), più che altro per una questione stilistica di uniformità con gli altri elementi. Agli attributi sopra elencati si aggiungono un id per l’identificazione degli album creati dell’utente stesso ed uno per le immagini caricate;
2. l’immagine avrà un id univoco altre che titolo, data di caricamento, descrizione e percorso in cui è memorizzata come attributi. A questi ho deciso di aggiungere l’id dell’utente che l’ha caricata per mantenere l’associazione utente-immagine;
3. l’album, oltre che il suo id univoco, sarà munita di elementi come titolo, data di creazione e id del creatore;
4. il commento, in fine, disporrà di un attributo id e testo. Inoltre, a questi si aggiungono l’id dell’utente che ha scritto il commento e l’id dell’immagine associata al commento.

Per quanto riguarda le pagine web che compongono l’applicazione abbiamo:

* login
* sign-up
* home page
* album page
* image page
* error page

L’ultima fra queste è stata aggiunta per mostrare all’utente eventuali errori di elaborazione dei dati inseriti dallo stesso oppure di elaborazione del server

Le specifiche relative al progetto HTML puro da tenere in considerazione sono soprattutto le seguenti:

1. lo username che deve essere unico nonostante la scelta personale di aggiungere un altro campo univoco (id);
2. possibilità di associare a più album la stessa immagine creando quindi una relazione “molti-molti”;
3. le immagini sono associate ad uno o più commenti che possono provenire anche da utenti diversi;
4. la tabella che mostra le immagini nella “album page” deve mostrarne sempre 5 slot a prescindere dal numero di immagini presenti nell’album. Se quindi sono meno di cinque i restanti spazi per arrivare al completamento vengono aggiunti vuoti;
5. lo scorrimento all’interno dell’album si fa a gruppi di cinque immagini;
6. il bottone di cancellazione dell’immagine deve comparire soltanto se l’utente è il creatore della stessa ed elimina non solo l’immagine, ma anche tutti i commenti ad essa associati.

Per quanto riguarda gli eventi da gestire nel progetto, dalla specifica ricaviamo:

* pulsante di login, che avvia il processo di validazione delle credenziali inserite dall’utente;
* bottone di registrazione, che, dopo aver controllato i dati, li registra nel database;
* la creazione di un nuovo album tramite opportuno form;
* il click su un album all’interno della “home page” che fa cambiare pagine mostrando la relativa “album page”;
* i pulsanti “Successive” e “Precedenti” che regolano lo scorrimento dell’album;
* l’aggiunta di un commento ad una immagine;
* la cancellazione dell’immagine;
* i bottoni che per mettono di tornare indietro ad un album, alla home page oppure il logout direttamente

## 1.3 Estensione JSP

La specifica generale trattata al punto 1.1 viene ampliata. La possibilità di usare Java Script permette ora di introdurre delle funzionalità che portano a rivisitare il progetto:

“Si realizzi un’applicazione client-server web che modifica le specifiche precedenti come segue:

* L’applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione richiede l’inserimento di username, indirizzo di email e password e controlla la validità sintattica dell’indirizzo di email e l’uguaglianza tra i campi “password” e “ripeti password”, anche a lato client. La registrazione controlla l’unicità dello username.
* Dopo il login dell’utente, l’intera applicazione è realizzata con un’unica pagina.
* Ogni interazione dell’utente è gestita senza ricaricare completamente la pagina, ma produce l’invocazione asincrona del server e l’eventuale modifica del contenuto da aggiornare a seguito dell’evento.
* L’evento di visualizzazione del blocco precedente/successivo d’immagini di un album è gestito a lato client senza generare una richiesta al server. L’applicazione carica le informazioni necessarie per la visualizzazione di tutte le immagini di un album e dei relativi commenti mediante un’unica chiamata.
* Quando l’utente passa con il mouse su una miniatura, l’applicazione mostra una finestra modale con tutte le informazioni dell’immagine, tra cui la stessa a grandezza naturale, i commenti eventualmente presenti e la form per inserire un commento.
* L’applicazione controlla anche a lato client che non si invii un commento vuoto.
* Errori a lato server devono essere segnalati mediante un messaggio di allerta all’interno della pagina.
* Si deve consentire all’utente di riordinare l’elenco delle immagini all’interno di un album con un criterio personalizzato diverso da quello di default (data decrescente). L’utente accede a un elenco dei titoli delle immagini ordinato secondo il criterio correntemente in uso: default o personalizzato. Trascina il titolo di un’immagine nell’elenco e lo colloca in una posizione diversa per realizzare l’ordinamento che desidera, senza invocare il server. Quando l’utente ha raggiunto l’ordinamento desiderato, usa un bottone “salva ordinamento”, per memorizzare la sequenza sul server. Ai successivi accessi, l’ordinamento personalizzato, per quell’utente, è usato al posto di quello di default.”

## 1.4 Analisi specifiche JSP

Come possiamo notare le specifiche precedenti della versione HTML pura sono state revisionate, in particolare viene modificato il comportamento di:

registrazione e login che adesso devono avvenire in un’unica pagina;

gestione delle pagine, inquanto, dopo il login, l’applicazione dovrà essere realizzata in un’unica pagina;

visualizzazione delle immagini in un album. Queste non vengono più richieste al server di volta in volta a gruppi di cinque, bensì una solo volta tutte insieme. Sarà poi il lato client tramite JS ad occuparsi della visione in blocchi;

della pagina “image page” che non solo diventa una finestra modale, ma le informazioni da mostrare le vengono passate dalla pagina “album page”, che le ha già caricate preventivamente, piuttosto che dal server;

agli errori non viene più dedicata una pagina separata ma un “alert” all’interno del sito.

I nuovi elementi a cui prestare particolare attenzione, introdotti in questo “aggiornamento” del progetto, sono:

il controllo anche lato client degli elementi inseriti dall’utente nel form di registrazione;

l’aggiornamento parziale, di solo alcuni elementi della pagina, in base all’interazione dell’utente;

la finestra modale per mostrare quella che prima era la “image page”, da chiamare solo in caso di evento “mouseover”;

il controllo preventivo lato client per commenti vuoti;

il nuovo evento di riordinamento arbitrario dell’ordine per le immagini presenti nell’album;

l’introduzione della funzionalità “drag and drop” per gestire l’ordinamento sopra citato;

la memorizzazione nel database del nuovo ordinamento in modo da poterlo annoverare nelle future interazioni dell’utente.

## 1.5 Scelte progettuali

### 1.5.1 Versione generale

Per la progettazione della galleria di immagini, partendo dalla documentazione fornita, ho deciso di prendere delle decisioni in merito ad alcuni punti a mio avviso poco chiari o lasciati vaghi:

* lo username non può contenere caratteri speciali per essere valido;
* ad ogni utente viene assegnato al momento della registrazione un album chiamato “allPhotos” contenente tutte le immagini caricate dall’utente stesso. Questo album sarà sempre il primo a comparire nella lista degli album personali dell’utente. Inoltre, è l’unico album che può essere creato vuoto in quanto al momento della registrazione non sono ovviamente presenti immagini da inserire al suo interno;
* la creazione di un album prevede l’obbligo di selezionare almeno una foto da aggiungere allo stesso;
* immagini ed album possono avere tra loro lo stesso nome. Sarà dunque possibile creare album con lo stesso nome, così come dare titoli uguali a più immagini anche dello stesso utente
* la funzionalità di logout è disponibile solo nella Home Page;
* la cancellazione di una immagine, a prescindere dall’album che la contiene, la rimuove in automaticamente da ogni album in cui è presente. L’utente viene reindirizzato alla Home Page

### 1.5.2 Versione RIA

Oltre alle scelte precedentemente elencate si aggiungono quelle relative alle nuove funzionalità JS:

* l’ordinamento delle immagini nell’album non è immediatamente visibile, ma sarà disponibile dal successivo accesso allo stesso;
* l’aggiunta di un commento nella finestra modale “Image Page” crea un pop-up nella stessa che mostra quindi un form aggiornato contenente anche il nuovo commento;
* in caso di errore in Image Page viene mostrato un pop-up che lo mostra senza chiudere la finestra ;
* la chiusura della finestra modale provoca il ricaricamento della intera pagina Album Page
* la cancellazione dell’immagine porta alla chiusura di Image Page

# 2 Progettazione base di dati

## 2.1 Schema ERImmagine che contiene diagramma, linea, Disegno tecnico, Piano Descrizione generata automaticamente

## 2.2 Struttura del database

### 2.2.1 User table

CREATE TABLE `User` (

`id` INT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`email` VARCHAR(100) NOT NULL,

`password` VARCHAR(10) NOT NULL,

`username` VARCHAR(20) NOT NULL,

UNIQUE INDEX `email\_UNIQUE` (`email` ASC) VISIBLE,

UNIQUE INDEX `username\_UNIQUE` (`username` ASC) VISIBLE

)

### 2.2.2 Album table

CREATE TABLE `Album` (

`id` INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`title` VARCHAR(45) NOT NULL,

`creationDate` DATETIME NOT NULL,

`userId` INT NOT NULL,

FOREIGN KEY(userId) references User (id) on DELETE cascade

)

### 2.2.3 Image table

CREATE TABLE `Image` (

`id` INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`title` VARCHAR(45) NOT NULL,

`creationDate` DATETIME NOT NULL,

`description` VARCHAR(255) NOT NULL,

`path` VARCHAR(255) NOT NULL,

`userId` INT NOT NULL,

FOREIGN KEY(userId) references User (id) on DELETE cascade

)

### 2.2.3 ImageAlbumLink table

CREATE TABLE `ImageAlbumLink` (

`chosenOrder` INT NOT NULL,

`albumId` INT NOT NULL,

`imageId` INT NOT NULL,

PRIMARY KEY (`albumId`, `imageId`)

FOREIGN KEY(albumId) references Album (id) on DELETE cascade,

FOREIGN KEY(imageId) references Image (id) on DELETE cascade

)

### 2.2.4 Comment table

CREATE TABLE ‘Comment` (

`id` INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`text` VARCHAR(155) NOT NULL,

`imageId` INT NOT NULL,

`userId` INT NOT NULL,

FOREIGN KEY(userId) references User (id) on DELETE cascade,

FOREIGN KEY(imageId) references Image (id) on DELETE cascade

)

# 3 Versione HTML pura

## 3.1 Rappresentazione IFML

Immagine che contiene testo, diagramma, Piano, schematico

Descrizione generata automaticamente

## 3.2 Struttura codice

### 3.2.1 Model objects (beans)

Le classi che si occupano di rappresentare gli oggetti contenuti nel database sono:

* User
* Album
* Image
* ImageAlbumLink
* Comment

### 3.2.2 Data Access Objects (DAO)

Le classi che si occupano di comunicare con il database sono:

* UserDAO
* AlbumDAO
* ImageDAO
* ImageAlbumLinkDAO
* CommentDAO

### 3.2.3 Controller (Servlets)

Le classi che si occupano di mettere in comunicazione client e server per lo scambio di informazioni e il popolamento delle view sono:

* AddComment
* CreateNewAlbum
* DeleteImage
* GoToAlbumPage
* GoToHomePage
* GoToImagePage
* GoToLogin
* GoToSignUp
* Login
* Logout
* SignUp
* UploadImage

### 3.2.4 Filters

* LoginCheck

### 3.2.4 Views

Le pagine web vere e proprie viste dal client sono:

* albumPage.html
* error.html
* homepage.html
* imagePage.html
* infoPane.html
* login.html
* signUp.html

## 3.3 Sequence diagrams

### 3.3.1 Gestione cambio pagine

È importante far notare fin da subito che in caso di qualsiasi tipo di errore, sia che si tratti di un problema dovuto all’input inserito o di un errore lato server, l’utente verrà reindirizzato alla schermata “error.html”. questa mostrerà un messaggio personalizzato per indicare all’utente la causa dell’errore.

La pagina html in questione mostra però due aspetti:

se l’utente non è registrato al momento dell’errore viene mostrata la pura pagina “error.html” che offre solo la possibilità di tornare al login;

in caso sia presente un utente in sessione la pagina di errore viene arricchita tramite. “infoPane.html” che mostra una schermata più personalizzata con info relative all’utente specifico. In questo caso l’utente avrà a disposizione due opzioni: tornare alla “Home Page” oppure effettuare il logout

#### Immagine che contiene testo, diagramma, Parallelo, numero Descrizione generata automaticamente3.3.1.1 Comportamento login

La schermata della “Login Page” è il file html su cui l’utente viene indirizzato all’apertura del sito web. A seguito del corretto inserimento dei dati personali richiesti l’utente viene automaticamente reindirizzato alla schermata “homePage.html”. Quest’ultima viene preparata dal server con tutte le informazioni necessarie per l’utente in questione tramite la servlet “GoToHomePage”

#### Immagine che contiene testo, diagramma, Parallelo, linea Descrizione generata automaticamente3.3.1.2 Comportamento registrazione nuovo utente

La schermata signUp.html può essere chiamata cliccando il pulsante “SignUp” presente nella schermata “login.html”. A seguito dell’apertura della pagina dedicata alla registrazione, l’utente è richiesto inserire i suoi dati personali e, a seguito di un controllo di validità di questi ultimi lato server, l’utente è reindirizzato alla “Login Page” da dove potrà accedere usando il suo nuovo account

#### Immagine che contiene testo, diagramma, Parallelo, numero Descrizione generata automaticamente3.3.1.3 Selezione di un album dalla Home Page

#### Immagine che contiene testo, diagramma, linea, Parallelo Descrizione generata automaticamente3.3.1.4 Selezione di una immagine dall’Album Page

### 3.3.2 Gestione eventi nelle singole pagine

#### 3.3.2.1 Eventi disponibili dalla Home Page

Immagine che contiene testo, linea, diagramma, schermata

Descrizione generata automaticamente

Immagine che contiene testo, diagramma, Parallelo, linea

Descrizione generata automaticamente

Immagine che contiene testo, diagramma, linea, Parallelo

Descrizione generata automaticamente

#### 3.3.2.2 Eventi disponibili dall’Album Page

Immagine che contiene testo, linea, schermata, diagramma

Descrizione generata automaticamente

Immagine che contiene testo, diagramma, Parallelo, numero

Descrizione generata automaticamente

#### 3.3.2.3 Eventi disponibili dalla Image PageImmagine che contiene testo, linea, schermata, diagramma Descrizione generata automaticamenteImmagine che contiene testo, schermata, linea, Carattere Descrizione generata automaticamente

Immagine che contiene testo, diagramma, Parallelo, linea

Descrizione generata automaticamente

Immagine che contiene testo, diagramma, linea, Parallelo

Descrizione generata automaticamente

# 4 Versione RIA

## 4.1 Rappresentazione IFML

Immagine che contiene diagramma, Piano, Disegno tecnico, schematico

Descrizione generata automaticamente

## 4.2 Struttura codice

### 4.2.1 Model objects (beans)

Le classi che si occupano di rappresentare gli oggetti contenuti nel database sono:

* User
* Album
* Image
* ImageAlbumLink
* Comment

### 4.2.2 Data Access Objects (DAO)

Le classi che si occupano di comunicare con il database sono:

* UserDAO
* AlbumDAO
* ImageDAO
* ImageAlbumLinkDAO
* CommentDAO

### 4.2.3 Controller (Servlets)

Le classi che si occupano di mettere in comunicazione client e server per lo scambio di informazioni e il popolamento delle view sono:

* AddComment
* CreateNewAlbum
* DeleteImage
* GoToAlbumPage
* GoToHomePage
* GoToLogin
* GoToSignUp
* Login
* Logout
* SignUp
* UploadImageOrder
* UploadImage

Rispetto alla versione HTML pura notiamo l’aggiunta della servlet “UploadImageOrder” che si occupa di cambiare l’ordinamento delle immagini in un album e la rimozione di “GoToImagePage” in quanto non necessaria passando tutte le informazioni da mostrare in Image Page direttamente dall’Album Page

### 4.2.4 Filters

* LoginCheck
* NoCacher

### 4.2.5 Views

Le pagine web vere e proprie viste dal client sono solamente due in questa versione:

* login.html -> dedicate al login e alla registrazione dell’utente
* home.html -> contiene all’interno tutte le precedenti pagine (home, album e image) che vengono oscurate o mostrate all’utente, al momento del bisogno, tramite dei divisori gestiti con Java Script

## 4.3 Sequence diagrams

### 4.3.1 Gestione cambio schermate

#### 4.3.1.1 Comportamento login

Immagine che contiene testo, Parallelo, diagramma, ricevuta

Descrizione generata automaticamente

#### 4.3.1.2 Comportamento registrazione nuovo utente

Immagine che contiene testo, diagramma, Parallelo, numero

Descrizione generata automaticamente

#### 4.3.1.3 Selezione di un album dalla Home Page

Immagine che contiene diagramma, testo, Parallelo, Piano

Descrizione generata automaticamente

#### 4.3.1.4 “Mouseover” su un’immagine dall’Album Page

Immagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamente

#### 4.3.1.5 “Mouseleave” dalla finestra modale Image Page

Immagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamente

### 4.3.2 Gestione eventi

#### 4.3.2.1 Eventi disponibili dalla Home Page

Immagine che contiene testo, schermata, diagramma, linea

Descrizione generata automaticamente

Immagine che contiene testo, diagramma, Parallelo, numero

Descrizione generata automaticamente

Immagine che contiene testo, diagramma, Parallelo, numero

Descrizione generata automaticamente

#### Immagine che contiene testo, schermata, linea, diagramma Descrizione generata automaticamenteImmagine che contiene testo, diagramma, linea, schermata Descrizione generata automaticamente4.3.2.2 Eventi disponibili dall’Album Page

Immagine che contiene testo, diagramma, Parallelo, numero

Descrizione generata automaticamente

#### 4.3.2.3 Eventi disponibili dall’Image Page

Immagine che contiene testo, diagramma, Parallelo, linea

Descrizione generata automaticamente

Immagine che contiene testo, diagramma, Parallelo, linea

Descrizione generata automaticamente