

A.A. 2015-2016

# Progetto Object Oriented Programming

Università degli Studi dell'Aquila

Versione preliminare del design object oriented.

# Indice dei contenuti di questo deliverable:

---

## **1. Requirements collection**

*1.1: Functional requirements*

*1.2: Non functional requirements*

*1.3: Assumptions*

## **2. System design**

*2.1: Design decisions*

*2.2: Use case diagram*

*2.3: Use case description*

## **3. Software Architecture**

*3.1: Component diagram*

*3.2: Component diagram description*

## **4. Analysis Model**

*4.1 Robustness diagram*

*4.2 Class diagram*

# 1. REQUIREMENTS COLLECTION

---

## 1.1: Functional requirements

---

### 1.1.a: Business-Logic requirements:

Il sistema deve essere in grado di organizzare e gestire un portale per la trascrizione di antichi manoscritti, nel dettaglio deve:

- Fornire funzionalità per memorizzare immagini di grandi dimensioni provenienti dalle scansioni.
- Fornire una lista indicizzata di opere contenenti diversi tipi di dati.
- Fornire una GUI che permetta ai diversi tipi di utente di accedere con semplicità alle funzioni di cui ha bisogno.
- Garantire la coerenza dei dati mostrati all'utente e quelli nella struttura dati.
- Garantire la possibilità di "check" ad ogni passaggio (scansione, trascrizione, pubblicazione) da parte dei supervisori.

### 1.1.b: System Requirements:

Il nostro sistema dovrà avere una funzione di gestione, accessibile solamente agli admin, che gli consentirà di manipolare qualsiasi dato all'interno del sistema stesso.

## 1.2: Non functional requirements

### Safety:

Il committente non ha richiesto vincoli di safety, pertanto il sistema viene considerato non safety-critical. La safety sarà pertanto misurata solamente in base alla sicurezza dell'account di ogni utente.

**Usability:**

In base all'utenza che potrebbe usufruire del sistema, abbiamo deciso di propendere per un'interfaccia grafica estremamente user-friendly.

**Performance:**

Il sistema maneggia dati (soprattutto le immagini) molto pesanti, pertanto la performance potrebbe variare a seconda delle prestazioni degli end-system e dei collegamenti con le risorse.

**Operational:**

Andrà garantita coerenza nella sequenzialità dei dati relativi ad un'opera, nonché la corrispondenza tra opere ed indicizzazione.

## 1.3 Assumptions:

---

- Abbiamo assunto, usando MySQL, di poter evitare la gestione di accessi concorrenti al DB.
  - Abbiamo assunto che i collaboratori (es. trascrittori) abbiano accesso ai privilegi di un utente avanzato, oltre che a quelli che sono prerogative del loro settore.
  - Abbiamo assunto che la funzione di upload delle immagini nel web server si potrà effettuare dalla GUI tramite un apposito pulsante che permetterà di scegliere il file immagine dal proprio terminale.
  - Abbiamo assunto, a seguito dell'analisi della specifica, che i manoscritti dovranno essere digitalizzati per intero prima della messa in pubblicazione/specifica.
-

## 2. System Design

---

### 2.1 Design Decisions

---

**1-** Abbiamo optato per una struttura software locale che interagisce con un server ed un database online.

Questa scelta è stata adottata poiché abbiamo ritenuto più intuitiva e più performante una GUI realizzata su software locale rispetto ad altre soluzioni considerate. Un altro punto a favore è la compatibilità, infatti non dovendo dipendere da programmi esterni quali browser o altro, non ci saranno difficoltà di visualizzazione dovute ad utilizzo da piattaforme differenti.

**2 -** L'utente interagirà con una GUI che, a seconda del proprio login, offrirà determinate funzioni e interfacce.

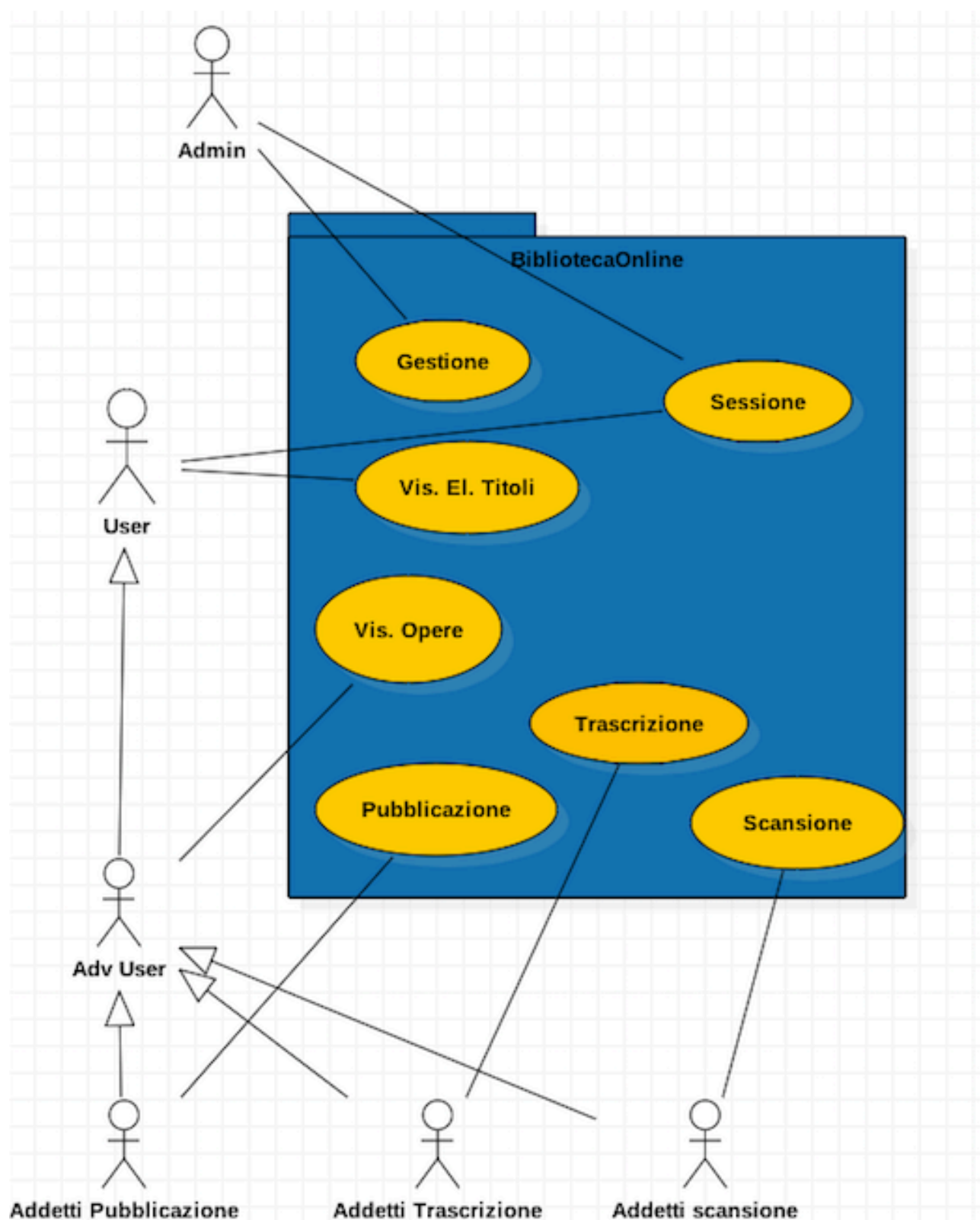
Scelta necessaria considerando che le categorie di utenze sono svariate e con funzioni, in alcuni casi, molto diverse tra loro.

**3 -** Il server conterrà esclusivamente i file multimediali, mentre il database MySQL conterrà riferimenti ad essi e le relative trascrizioni.

La decisione sopra citata viene naturale se si fa riferimento alle prestazioni del database e del web server. Disponendo MySQL di tipi di dati per la memorizzazione di testo sufficientemente ampi, abbiamo preferito la memorizzazione delle trascrizioni direttamente su DB piuttosto che su server.

## 2.2 Use Case diagram per il nostro sistema

---



## 2.3 Use Case description:

---

	<b>BibliotecaOnline( Package - visione generale del sistema)</b>
<b><u>Descrizione</u></b>	Offre le funzionalità necessarie al corretto funzionamento del sistema richiesto. I diversi tipi di utente avranno interazioni particolari con il sistema, per cui gli use case a seguire potranno essere esclusiva di una o più classi di utenti.
<b><u>Evento Scatenante</u></b>	Utilizzo della GUI.

	<b>System Requirement- Gestione</b>
<b><u>Descrizione</u></b>	Consente (esclusivamente agli amministratori) di manipolare diverse tipologie di dati all'interno del sistema.
<b><u>Evento Scatenante</u></b>	Login come admin con conseguente apertura del pannello di gestione.

	<b>Business-Logic Requirement - Sessione</b>
<b><u>Descrizione</u></b>	Consente di gestire le diverse tipologie di utenti, fornendo a ciascuno le funzioni di sistema appropriate.
<b><u>Evento Scatenante</u></b>	Login

	<b>Business-Logic Requirement- Vis. El. Titoli</b>
<b><u>Descrizione</u></b>	Funzionalità dell'utente base. Consente di visualizzare l'elenco dei titoli delle opere.
<b><u>Evento scatenante</u></b>	Click sull'apposito box della GUI.

	<b>Business-Logic Requirement- Vis. Opere</b>
<b><u>Descrizione</u></b>	Funzionalità dell'utente avanzato, consente di visualizzare un'intera opera, comprensiva di immagini ed eventualmente testo digitalizzato.
<b><u>Evento scatenante</u></b>	Dopo il login, selezione dell'opera tra la lista indicizzata dei titoli.

	<b>Business-Logic Requirement- Pubblicazione</b>
<b><u>Descrizione</u></b>	Consente di etichettare l'opera corrente come pubblicabile, da quel momento in poi gli utenti potranno visualizzarla.
<b><u>Evento scatenante</u></b>	Click dell'apposito checkbox.

	<b>Business-Logic Requirement- Trascrizione</b>
<b><u>Descrizione</u></b>	Tramite la schermata di trascrizione, gli utenti specializzati potranno avere a disposizione un editor di testo sulla destra della schermata e l'immagine del manoscritto sulla sinistra. Per i supervisor apparirà un box che consentirà di etichettare la trascrizione come accettata.
<b><u>Evento scatenante</u></b>	Accesso al textEditing button dalla GUI.

	<b>Business-Logic Requirement- Scansione</b>
<b><u>Descrizione</u></b>	Consente di fare l'upload delle immagini sul web server. In seguito all'accettazione da parte dei supervisor verranno pubblicate.
<b><u>Evento scatenante</u></b>	Accesso alla funzione upload della GUI.

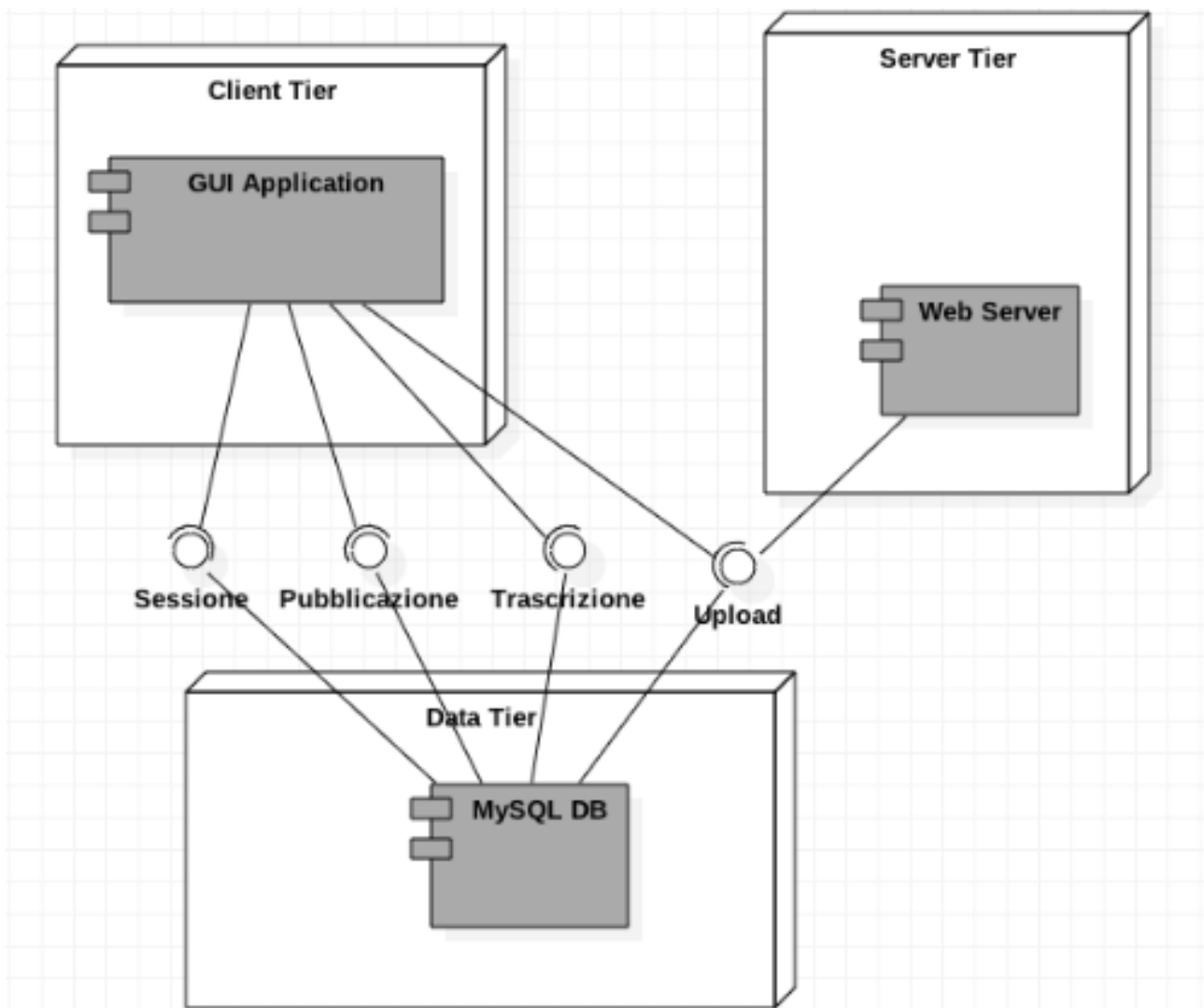


## 3. Software Architecture

---

### 3.1: Component Diagram per il nostro sistema:

---

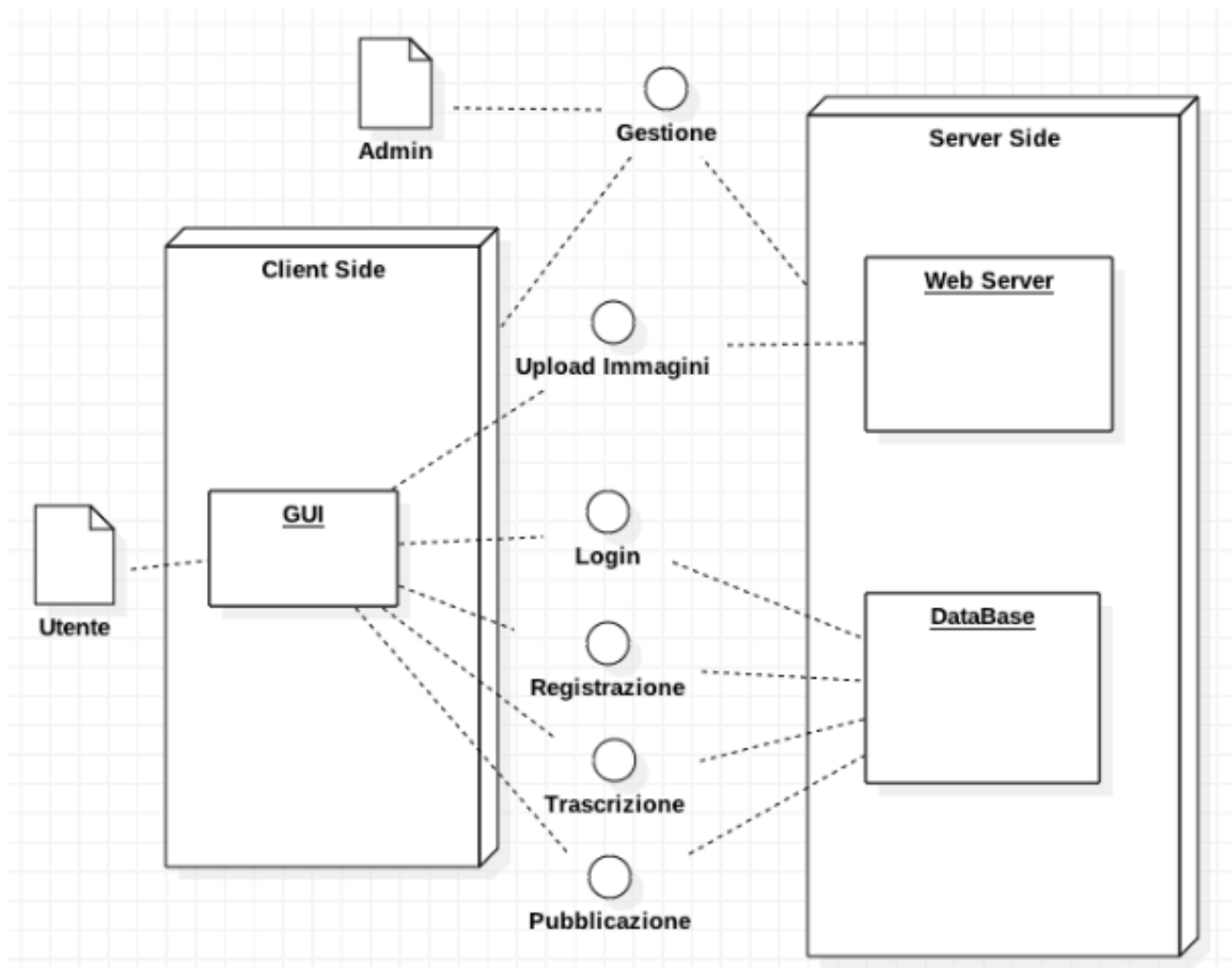


## 3.2 Component diagram description

---

Tutte le interazioni su web server e database passano attraverso le interfacce fornite dalla GUI. In dettaglio:

- Sessione  
Dovendo verificare le credenziali di autenticazione o crearne di nuove, deve interagire con il database.
- Pubblicazione  
Quando un'opera è pronta per la pubblicazione, viene utilizzata questa interfaccia che tramite una query su database ne modifica lo status.
- Trascrizione  
Nel nostro sistema le trascrizioni non fanno uso dello spazio su web server, motivo per cui questa interfaccia interagisce solo con il database.
- Upload  
Unica interfaccia che interagisce sia con web server che con database. Nonostante le immagini siano contenute nel web server, i riferimenti che le relazionano a pagine ed opere si trovano sul database.

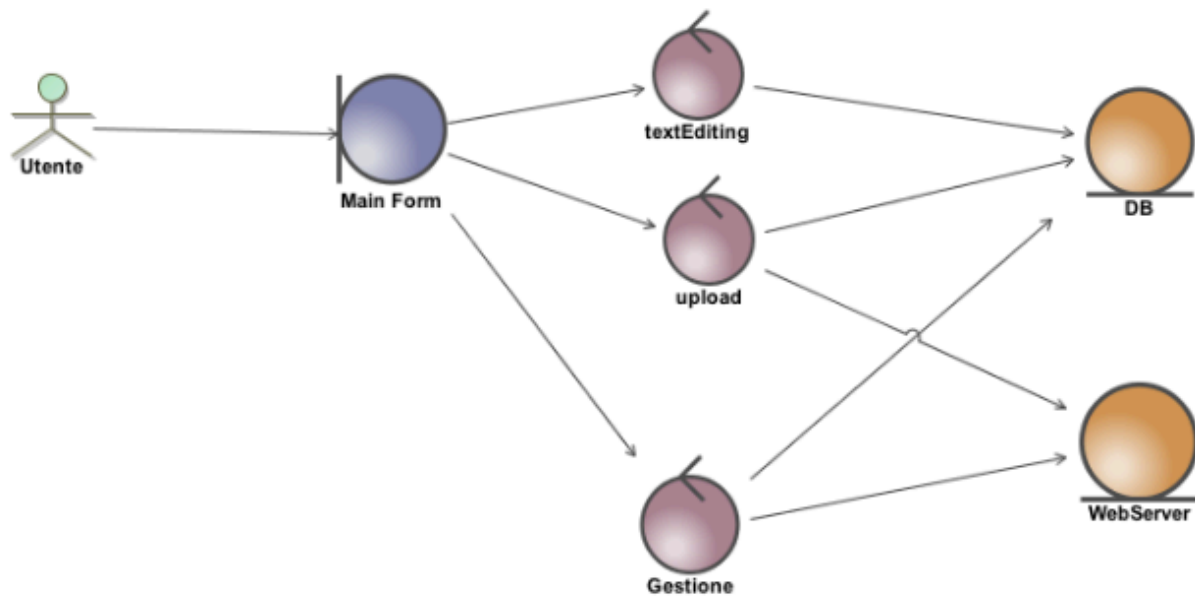


Vista dell'architectural design del nostro sistema, in cui si evidenzia la divisione client-server.

---

## 4. Analysis Model

---

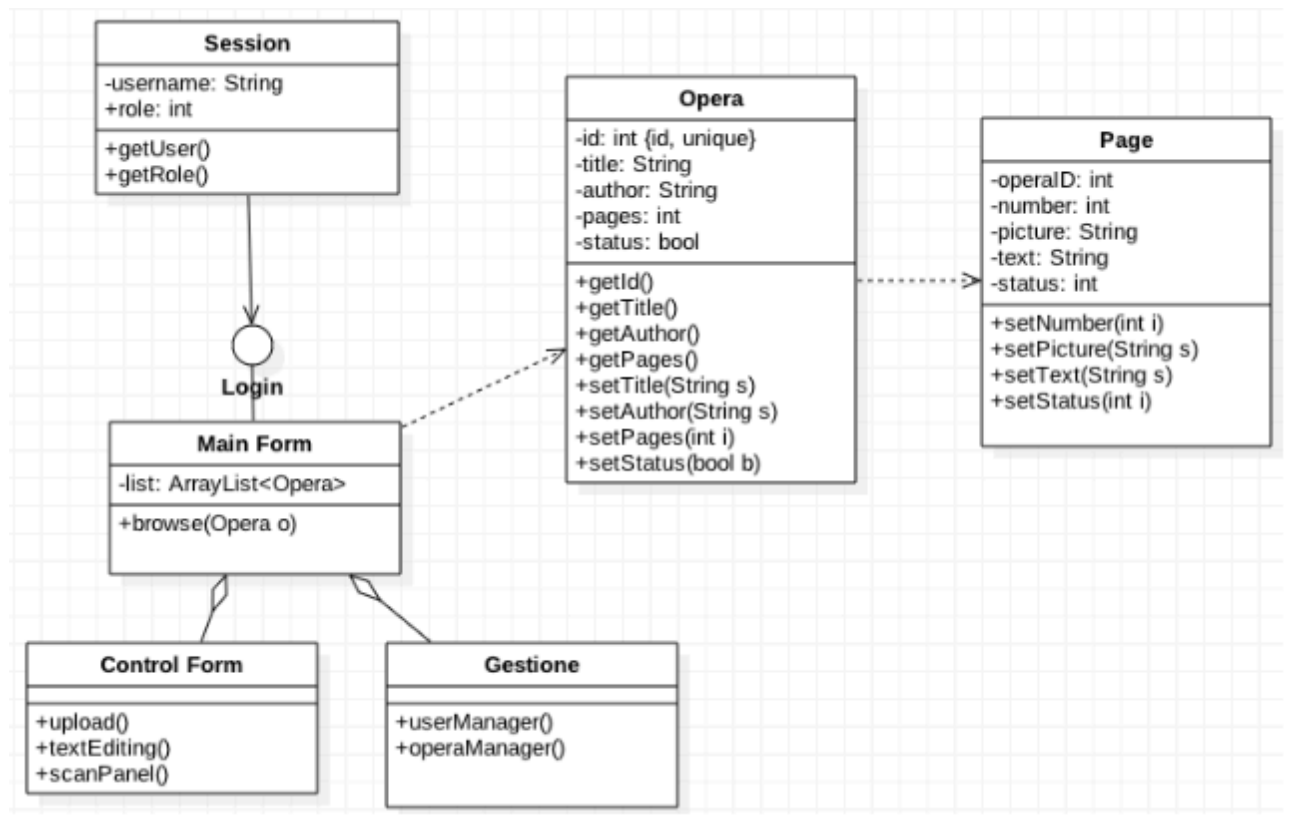


**4.1: Robustness diagram per il nostro sistema**

Abbiamo scelto la Main Form (GUI) come boundary item, dato che è l'entry point del sistema.

I controller textEditing, upload e Gestione, consentono agli utenti di accedere alle funzioni base del sistema e di operare sugli entity object DataBase e WebServer.

## 4.2 Class Diagram per il nostro sistema



- **Session**
  - "role" è un intero che cambia a seconda del ruolo, viene utilizzato dal programma per definire i permessi per quella tipologia di utenza.
- **Opera**
  - "id" è una sequenza di caratteri che caratterizzerà univocamente ogni opera. Ogni pagina sarà associata all'opera grazie a questa sequenza.
  - "status" è un booleano che definisce lo stato di pubblicazione dell'opera.

- **Page**

- "operaID" è la sequenza di caratteri che corrisponde all' "id" dell'opera a cui appartiene.
- "number" è il rispettivo numero di pagina.
- "picture" è il riferimento su web server dell'immagine acquisita per quella pagina.
- "status" è un int piuttosto che un bool perché la singola pagina può ritrovarsi in diversi stati: in attesa di scansione, attesa di accettazione della scansione, scansione rifiutata, attesa di trascrizione ecc. a differenza dell'opera che può semplicemente essere pubblicata/non pubblicata.

- **Main Form**

- "list" è la lista di tutte le opere presenti nel sistema, abbiamo scelto di utilizzare ArrayList perchè nel main form non subisce modifiche visto che ha uno scopo di visualizzazione ed è più efficiente a livello di memoria.
- "browse(Opera o)" data un'opera, questa funzione ne permette la visualizzazione in dettaglio.

- **Control Form**

- "textEditing()" apre l'editor e permette anche, agli utenti con i relativi privilegi, di revisionare la trascrizione.
- "scanPanel()" apre il pannello di gestione delle immagini scannerizzate e ne permette la revisione.

- **Gestione**

- "userManager()" e "operaManager()" sono le funzioni utilizzabili esclusivamente dall'amministratore del sistema e gli permettono di gestire gli utenti e le opere.

i

---

<sup>i</sup> Tutti i diagrammi nei formati originali si trovano nella cartella di GitHub.