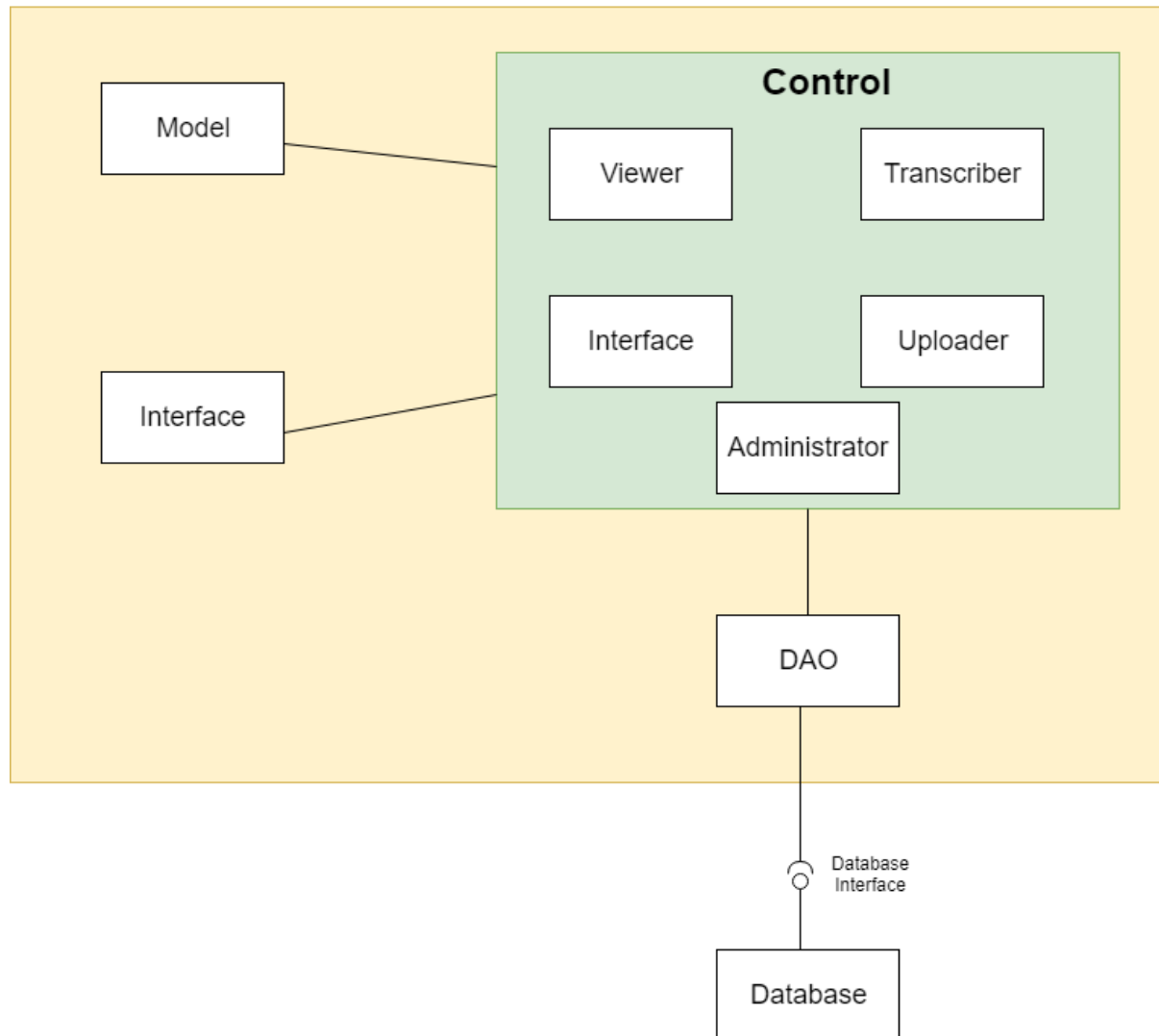


SYSTEM DESIGN

2.1 MODELLO ARCHITETTURA SOFTWARE

System



2.2 DESCRIZIONE DELL'ARCHITETTURA

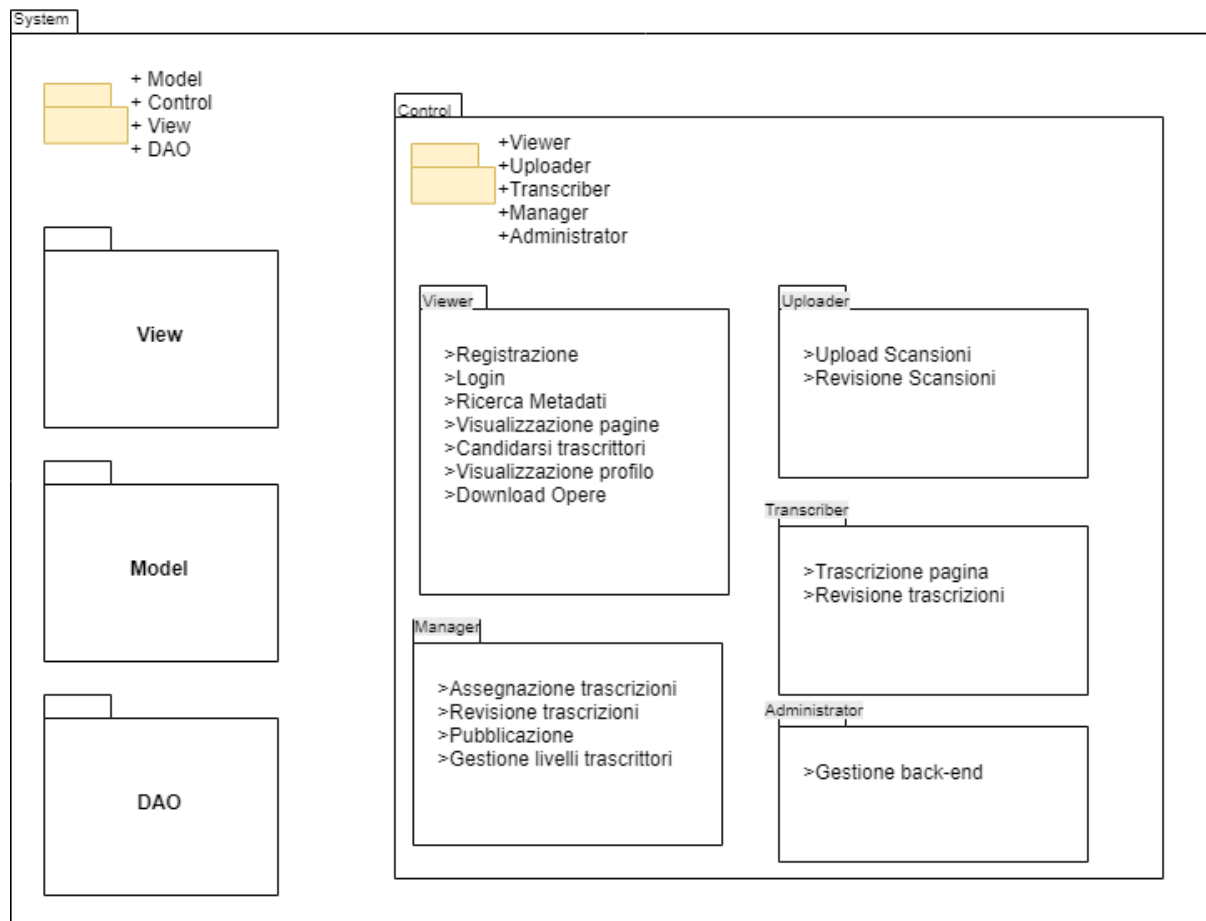
Il sistema è diviso in Interface, Controller, Database Manager e Model.

L'Interface è l'insieme delle classi che si occupano dell'User Interface, che hanno il compito di mediare tra l'utente e il Controller. Quest'ultimo si occupa di gestire tutte le funzionalità del sistema eseguendo i comandi richiesti attraverso l'interazione dell'Utente con l'Interfaccia, similmente ai Control evidenziati nell'analisi di Entity, Boundary e Control. (1.4). In particolare, il Control è diviso in cinque sottosistemi: Viewer, Transcriber, Manager, Uploader e Administrator, già specificati nell'Analisi dei Requisiti (1.1) e nel modello UML Use Case(1.2). Si tratta solo di una divisione logica delle classi, in base alle funzionalità che andranno ad implementare.

Il Database è gestito dal DBMS MySQL e si interfaccia con il Database Manager (DAO), che viene utilizzato dal Controller per operazioni di lettura o scrittura sulla base di dati.

In ultimo, il Model contiene tutte le classi necessarie per il funzionamento del Controller: le classi cioè che sono necessarie per effettuare le funzionalità richieste e specificate nel diagramma UML Use Case (1.2).

Inoltre, per meglio spiegare il System design fornito, è utile il Package Diagram relativo:



2.3 DESCRIZIONE DELLE SCELTE E DELLE STRATEGIE ADOTTATE

Il sistema è stato progettato secondo i paradigmi della modellazione Object-Oriented, con la priorità di rendere il sistema modulare per futuri utilizzi e per una migliore manutenzione: per questo abbiamo deciso di esplicitare la divisione del Control nei cinque sottosistemi logici. Mantenere questa modularità potrebbe inoltre essere fondamentale per implementare, in futuro, nuove funzionalità.

Il DBMS MySQL è stato scelto perché copre completamente qualunque funzionalità richiesta dal nostro sistema, ed allo stesso tempo è intuitivo e facile da utilizzare e mantenere. Si tratta di un DBMS sicuramente adatto a progetti di queste dimensioni.

L'User Interface deve essere intuitivo e facile da usare per l'utente registrato che si limita agli use-case contenuti nel package Viewer. Questi sono gli utenti ai quali il sistema è destinato, per la condivisione e conoscenza di memorie storiche contenute nei manoscritti. Dall'Analisi dei Requisiti (1.1) è emerso che gli utenti destinatari dovrebbero essere studenti dell'Università degli Studi dell'Aquila. La priorità dell'interfaccia per questi utenti dovrebbe quindi essere la chiarezza, sia nella navigazione sia nella consultazione delle pagine.

L'interfaccia per altri attori (Trascrittori, Uploader, Manager e Administrator), sebbene mantenendo la chiarezza e l'intuitività, deve presentare ulteriori videate e boundaries per permettere l'utilizzo degli use case richiesti dal corrispettivo package, o anche la loro assenza. Ad esempio, un utente troverà il pulsante "Candidati come Trascrittore" nel proprio profilo personale, mentre non sarà disponibile per un utente classificato come Trascrittore.

Il sistema è stato sviluppato concentrandosi sulle funzionalità, tenendo a mente che l'Administrator può cambiare il ruolo e i permessi di ogni attore in qualsiasi momento.