**3.1 Panoramica**

Il sistema vuole rendere più autonoma la prenotazione di un’aula per il docente e la ricerca di aule libere in maniera semplice e rapida.

Per far ciò il sistema offre due funzioni fondamentali:

● consente la visualizzazione di aule libere facilitando la ricerca allo studente;

● rende possibile la richiesta di prenotazione di una specifica aula direttamente al docente.

Gestione Prenotazioni quindi nasce con l’obiettivo facilitare l’organizzazione delle prenotazioni ed avere un quadro più chiaro di tutte le aule prenotate o libere.

Per la progettazione e per lo sviluppo di Gestione prenotazioni abbiamo scelto l’architettura MVC.



**3.2 Decomposizione sottosistemi**

Presentiamo una decomposizione del nostro sistemi in 3 layer (o livelli) che si occupano di gestire funzionalità ed aspetti differenti del nostro sistema:

● **Presentation Layer(View):** Livello che gestisce la comunicazione con le entità esterne al sistema tramite delle componenti che si occupano di rappresentare l’informazione verso i client, consentendo essi di interagire con il sistema.

●**Utente non registrato:** Gui che gestisce l’interfaccia degli utenti che non hanno effettuato l’accesso e/o non hanno effettuato la registrazione alla piattaforma, i quali sono limitati alla visualizzazione delle aule libere.

● **Studente/Docente**: Gui che gestisce l’interfaccia degli utenti che sono registrati e che hanno effettuato l’accesso alla piattaforma Gestione Prenotazioni, permette: di visualizzare gli orari delle aule libere, il profilo personale dell’utente e l’interfaccia riguardante la richiesta di prenotazione di un’aula.

● **Amministratore di dipartimento**: Gui che gestisce l’interfaccia dedicata al responsabile delle prenotazioni del dipartimento assegnato e permette di inserire o eliminare una prenotazione eseguita da un docente/studente.

●**Amministratore generale:** Gui che gestisce l’interfaccia dedicata al responsabile dell’ateneo e permette di creare o eliminare aule edifici e dipartimenti, inoltre, può assegnare le funzionalità di amministratore ad un docente, di conseguenza assegnare un amministratore ad un dipartimento.

**● Application Logic Layer(Control):** E’ il livello del sistema che si occupa del processamento dei dati necessario per produrre i risultati da inoltrare al

Presentation Layer.

**Gestione Utente**: Sottosistema che permette di registrarsi, effettuare login, logout, accedere ai servizi relativi all’area personale e la visualizzazione di quest’ultima.

**Gestioni Eventi Amm. di Dipartimento:** Sottosistema che permette la gestione di tutte le funzionalità di un amministratore di dipartimento, quali, accettazione o rifiuto di richieste di prenotazione, la prenotazione di un aula da parte di un amministratore e l’eliminazione di una prenotazione già accettata.

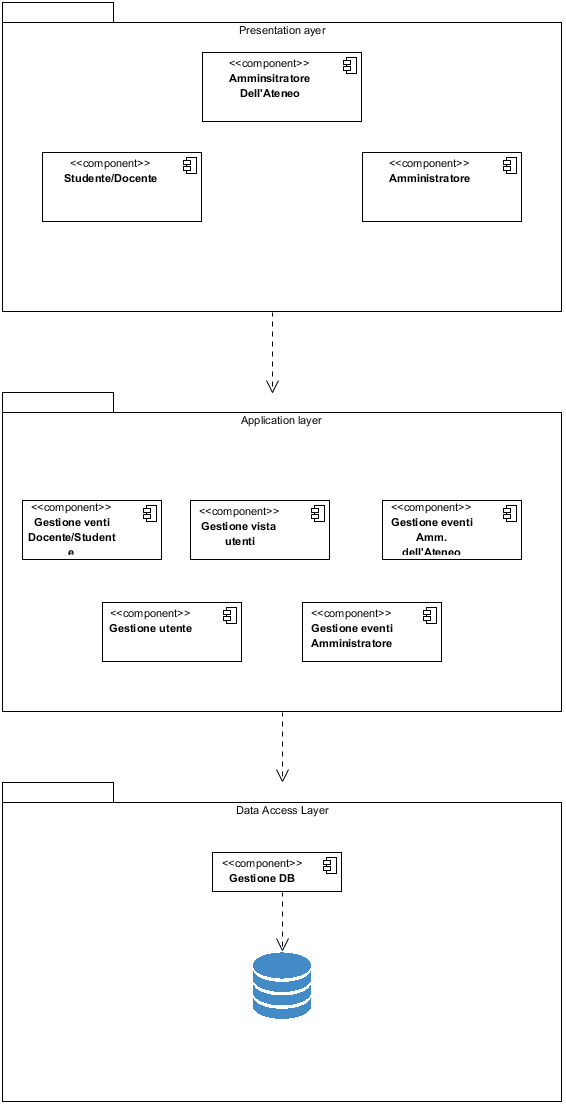
**Gestione vista degli utenti:**Questo sottosistema permette la visualizzazione degli edifici, aule, storico prenotazioni per utente, calendario e dipartimento.

**Gestione eventi Docente/Studente:** Sottosistema che permette la richiesta di prenotazione di un aula.

**Gestione eventi Amm. Generale:** Sottosistema che permette la gestione delle funzionalità dell’amministratore generale quali: aggiunta e rimozione di dipartimenti, edifici ed aule, l’assegnazione degli amministratori ai dipartimenti e la suddetta carica di amministratore ai docenti.

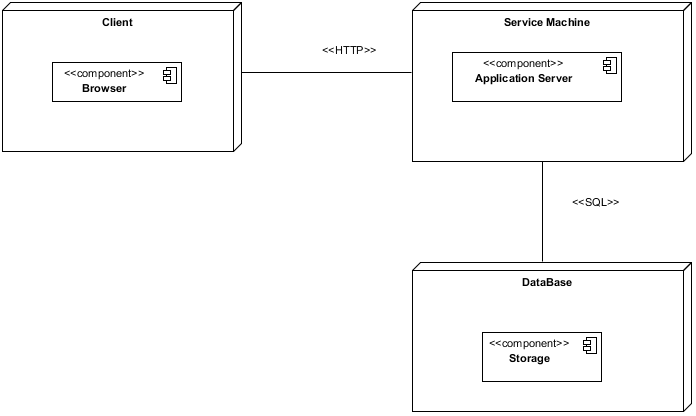
● **Data Access Layer(Model) :** Livello che gestisce i dati che sono necessari al funzionamento dell’intero sistema, ovvero i dati persistenti.

**Gestione DataBase**: sottosistema che si occupa di immagazzinare e di prelevare i dati persistenti dal nostro database.



**3.2.3 Diagramma di Deployment**

Gli utenti potranno interagire con Gestione prenotazioni tramite un WebBrowser collegandosi all’indirizzo del nostro sito, il Control si occupa di interpretare gli eventi generati dall’utente, richiedere e prelevare le opportune risorse dalla Base di Dati ed inviare le risposte tramite protocollo HTTP.



**3.3 Mapping hardware/software**

Per il progetto software che si vuole realizzare faremo uso di componenti “off-the-shelf”, che sono componenti software disponibili sul mercato per facilitare la creazione del progetto.

Per il sistema che si vuole realizzare ci interessa un framework o librerie per applicazioni web.

Il framework che andremo ad utilizzare è Bootstrap, jQuery, Javascript ed AJAX.

Bootstrap è un framework open source che contiene una raccolta di strumenti per facilitare lo sviluppo di interfacce web.

jQuery è una libreria JavaScript per accelerare lo sviluppo della business logic delle interfacce web.

Javascript è un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client.

Per il sistema basato su un’architettura MVC, sono stati scelti questi tipi di configurazioni:

per la parte di Interface (View) usiamo le JSP, HTML, CSS, JavaScript.

per la parte di business logic (Control) usiamo Apache Tomcat e Java.

per la parte di Data access (Model) usiamo MySQL con JDBC.

Come Protocollo di comunicazione usiamo HTTP, TCP/IP.

**3.4 Gestione dati persistenti**

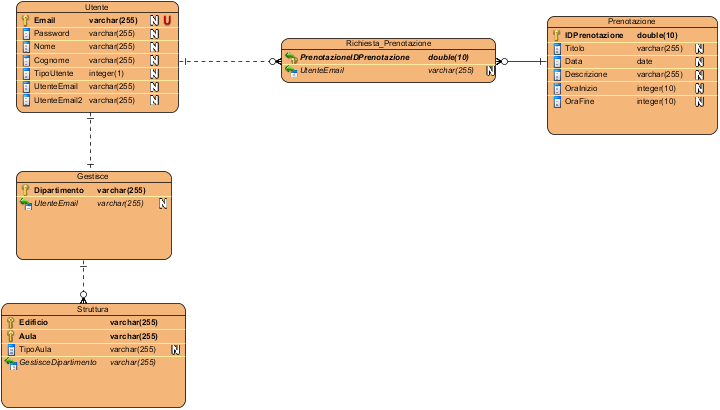
Il sistema che si vuole sviluppare sarà installato su un solo computer e utilizzerà un

DBMS MySQL installato sullo stesso. Il sistema sarà diviso in client e server che

inizialmente saranno lo stesso pc, ma in caso di futura necessità potranno essere

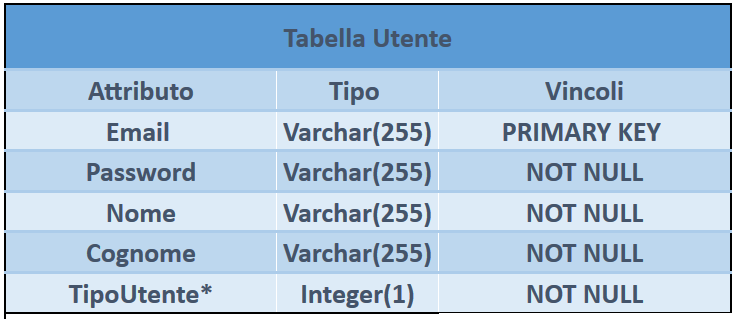
facilmente divisi in quanto i servizi saranno progettati separatamente. Sul server ci sarà un DBMS per la gestione dei dati persistenti di nostro interesse.

3.4.1 schema logico



3.4.2 Struttura delle Tabelle

Tabella utente



\* 0= Studente 1= Docente 2=Amm. di dipartimento 3= Amm. dell’Ateneo.

Tabella Richiesta Prenotazione

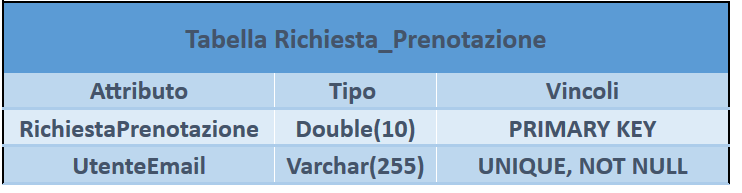


Tabella Prenotazione

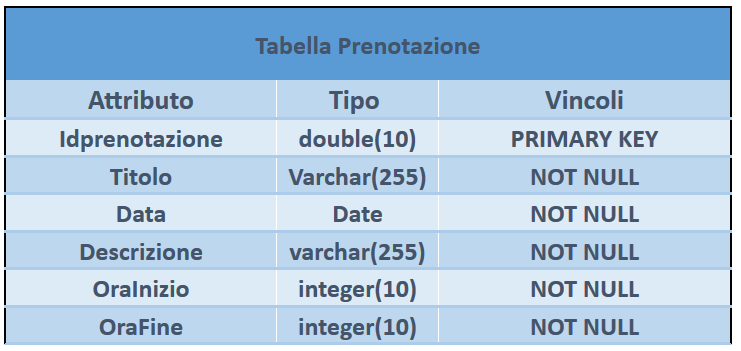


Tabella Gestisce

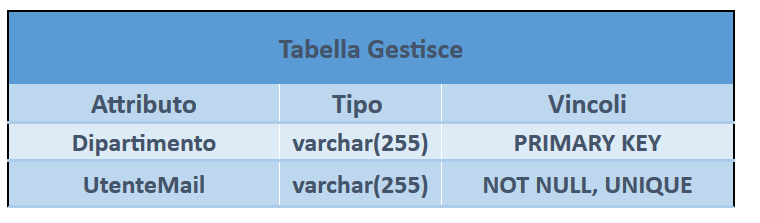
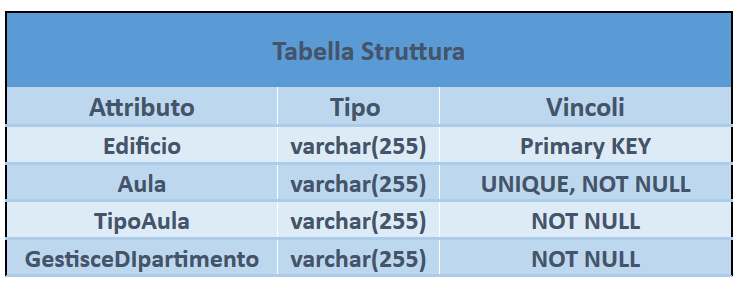


Tabella Struttura



**3.5 Controllo degli accessi e sicurezza**

L’accesso agli utenti è garantito dall’utilizzo di e-mail e password e verrà chiesto a ogni scadenza di sessione.

La sicurezza sui dati sensibili è garantita dall’accesso controllato, in quanto soltanto il gestore può avere accesso ai dati relativi agli utenti.

**3.6 Controllo globale del software**

Il controllo del flusso del software è regolato da classi Java che fungono da ricevitori di eventi e che rispondono alle attivazioni dei client.

Le richieste vengono generate da un client e la classe preposta a gestire quel determinato evento associato alla richiesta, prendendo talvolta degli input, si preoccupa di inizializzare le richieste e di inoltrarle alle classi per lo svolgimento dell’operazione. Una volta ottenuto il risultato dell’operazione, la classe gestore

si preoccupa di inoltrarlo al client che aveva generato la richiesta.

Il sistema software è gestito con l’uso di Servlet e JSP.

Il server centrale attende le richieste di un client (web browser) e una volta ricevuta una richiesta, la processa e la smista alla Servlet deputata.

**3.7 Condizione limite**