



Gestione Prenotazioni

---

# SDD System Design Document

Gestione Prenotazione

## TEAM MEMBER

Borriello Martina	0512105144
Claudini Aldo	0512105486
Claro Andrea	0512105252
Marcantuono Veronica	0512105426



## **Sommario**

<b>1. Introduzione .....</b>	<b>3</b>
<b>1.1 Obiettivi del sistema .....</b>	<b>3</b>
<b>1.2 Design Goals .....</b>	<b>3</b>
<b>1.3 Definizione, Acronimi e Abbreviazioni.....</b>	<b>5</b>
<b>1.4 Riferimenti .....</b>	<b>5</b>
<b>1.5 Panoramica.....</b>	<b>5</b>
<b>2. Architettura del Sistema corrente .....</b>	<b>6</b>
<b>3. Architettura del Sistema proposto .....</b>	<b>6</b>
<b>3.1 Panoramica .....</b>	<b>6</b>
<b>3.2 Decomposizione sottosistemi.....</b>	<b>7</b>
<b>3.3 Mapping hardware/software .....</b>	<b>10</b>
<b>3.3.1 Deployment.....</b>	<b>11</b>
<b>3.4 Gestione dati persistenti .....</b>	<b>11</b>
<b>3.4.1 Schema logico.....</b>	<b>12</b>
<b>3.4.2 Struttura delle Tabelle .....</b>	<b>13</b>
<b>3.5 Controllo degli accessi e sicurezza .....</b>	<b>15</b>
<b>3.6 Controllo globale del software.....</b>	<b>18</b>
<b>3.7 Condizione limite .....</b>	<b>19</b>
<b>4. Servizi dei Sottosistemi.....</b>	<b>20</b>
<b>5. Glossario .....</b>	<b>26</b>



## 1. Introduzione

### 1.1 Obiettivi del sistema

L'Università degli studi di Salerno attualmente ha una gestione delle prenotazioni delle aule che comporta un oneroso dispendio di tempo. L'obiettivo del progetto è la realizzazione di una piattaforma che automatizzi il servizio, semplificando e riducendo i tempi poiché in questo modo l'utente non deve rivolgersi personalmente ad un addetto. Inoltre, saranno offerte ulteriori funzionalità di riepilogo per tenere traccia delle prenotazioni effettuate e funzionalità di visualizzazione dell'Ateneo. Vista l'eterogeneità degli utenti, l'interfaccia del sistema dovrà essere estremamente intuitiva.

### 1.2 Design Goals

ID	DESCRIZIONE	CATEGORIA	ORIGINE
DG_1 Tempo di risposta	Il sistema deve garantire un tempo di risposta quanto più breve è possibile.	Prestazioni	NFR_10
DG_2 Portabilità	Il sistema sarà responsive, in quanto il sito sarà sviluppato in modo da permettere la visualizzazione anche da tablet e cellulare.	Usabilità Implementazione	NFR_2 NFR_17
DG_3 Usabilità	L'usabilità del sistema sarà data dall'interfaccia intuitiva, contenente pulsanti, titoli e messaggi di errore semplici da comprendere.	Usabilità	NFR_3
DG_4 Throughput	In caso di elevato numero di utenti attivi (massimo), la navigazione del sito non deve bloccarsi.	Prestazioni	NFR_10
DG_5 Sicurezza	Il sistema consente l'accesso tramite autenticazione ed ogni utente può usufruire esclusivamente delle funzionalità associate alla propria categoria. Inoltre, il sistema deve tenere al sicuro i dati da chi non è autorizzato ad accedervi.	Affidabilità	NFR_4, NFR_5, NFR_6



DG_6 Memoria	Si utilizzerà il database relazionale MySQL per gestire la grande quantità di dati, pertanto il sistema non dovrà subire rallentamenti.	Implementazione	NFR_16
DG_7 Disponibilità	Il sistema dovrà essere perennemente usufruibile dagli utenti, fatta eccezione per le ore di manutenzione.	Affidabilità	NFR_8
DG_8 Robustezza	Eventuali inserimenti non validi o mancanti da parte dell'utente devo essere segnalati tramite messaggi d'errore.	Affidabilità	NFR_7
DG_9 Affidabilità	Il sistema garantirà una corretta gestione delle funzionalità.	Affidabilità	NFR_4 NFR_5 NFR_6
DG_10 Adattabilità	Il sistema dovrà adattarsi facilmente alle nuove necessità del cliente.	Implementazione	NFR_18

### 1.3 Definizione, Acronimi e Abbreviazioni

**RAD:** Requirements Analysis Document.

**SDD:** System Design Document.

**ODD:** Object Design Document.

**NFR:** Requisiti non funzionali.

**UC:** Use case.

**FR:** Requisiti funzionali.

**DG:** Design Goal

**MVC:** Model View Controller

**GUI:** Graphic User Interface

**DBMS:** Database Management System

**JSP:** Java Server Page

**HTTP:** HyperText Transfer Protocol

**Amministratore:** Utente che si occupa di gestire le richieste di prenotazione.

**Amministratore Ateneo:** Utente che si occupa della gestione della struttura dell'Ateneo e di designare gli Amministratori.

**Utente:** Un qualsiasi utilizzatore della piattaforma, che sia esso Studente o Docente.

### 1.4 Riferimenti

- Object-Oriented Software Engineering Using UML, Patterns, and Java, Bernd Bruegge & Allen H. Dutoit.
- Easycourse.unisa.it
- Gestione Prenotazione RAD

### 1.5 Panoramica

Il documento si compone di una prima parte in cui vengono introdotti gli obiettivi di design.

Al secondo punto del documento viene presentata l'architettura del sistema corrente.

Al terzo punto viene presentata l'architettura del sistema proposto e nel dettaglio:

- La decomposizione del sistema in sottosistemi;
- Il mapping Hardware/Software;
- La gestione dei dati persistenti;
- Il controllo degli accessi e della sicurezza;
- Il controllo del flusso globale del sistema;
- Le condizioni limite.

Al quarto punto vengono presentati i servizi di ogni sottosistema.

Al quinto punto viene fornito il glossario dei termini utilizzati nel documento con le relative definizioni.

## 2. Architettura del Sistema corrente

Attualmente non esiste una piattaforma che dia la possibilità agli studenti ed ai docenti di poter prenotare un'aula.

Tutto questo avviene ancora tramite e-mail, cellulare o bisogna recarsi in segreteria fisicamente e tutto questo è possibile solo per i docenti.

Attualmente l'unica piattaforma che esiste permette solo la visualizzazione delle aule e il sistema attuale consiste in un database di aule libere con relativi orari ordinati in base agli edifici dell'università.

## 3. Architettura del Sistema proposto

### 3.1 Panoramica

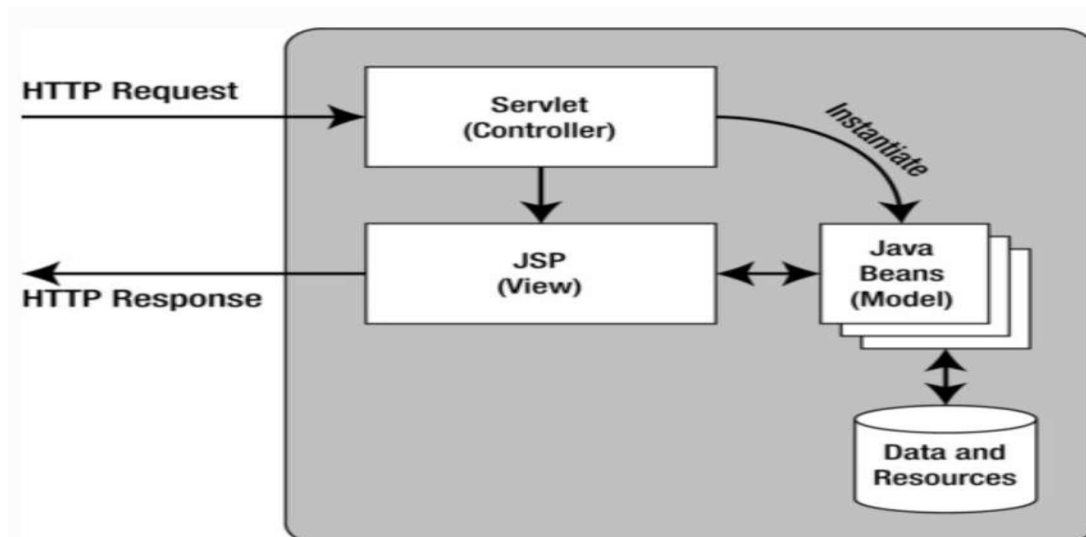
Il sistema proposto vuole rendere più autonoma la prenotazione di un'aula per il docente e in più dare anche allo studente la possibilità di poterla prenotare.

Per far ciò il sistema offre delle funzioni fondamentali:

- consente la visualizzazione di aule libere, facilitando la ricerca allo studente ed al docente.
- rende possibile la richiesta diretta di prenotazione delle aule dell'ateneo al docente.
- rende possibile la richiesta di prenotazione agli studenti di specifiche aule riservate a loro.

Gestione prenotazioni quindi nasce con l'obiettivo di rendere semplice e soprattutto 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(Model View Controller) che separa la logica di business da quella di presentazione.



**Model:** Si occupa di gestire i dati persistenti, contiene metodi di lettura e scrittura sul Database.

**View:** Si occupa di far visualizzare i dati all'utente e si occupa dell'interazione tra quest'ultimo ed il Sistema.

**Controller:** Si occupa della logica di controllo dell'applicazione interagendo con gli altri due componenti.

### 3.2 Decomposizione sottosistemi

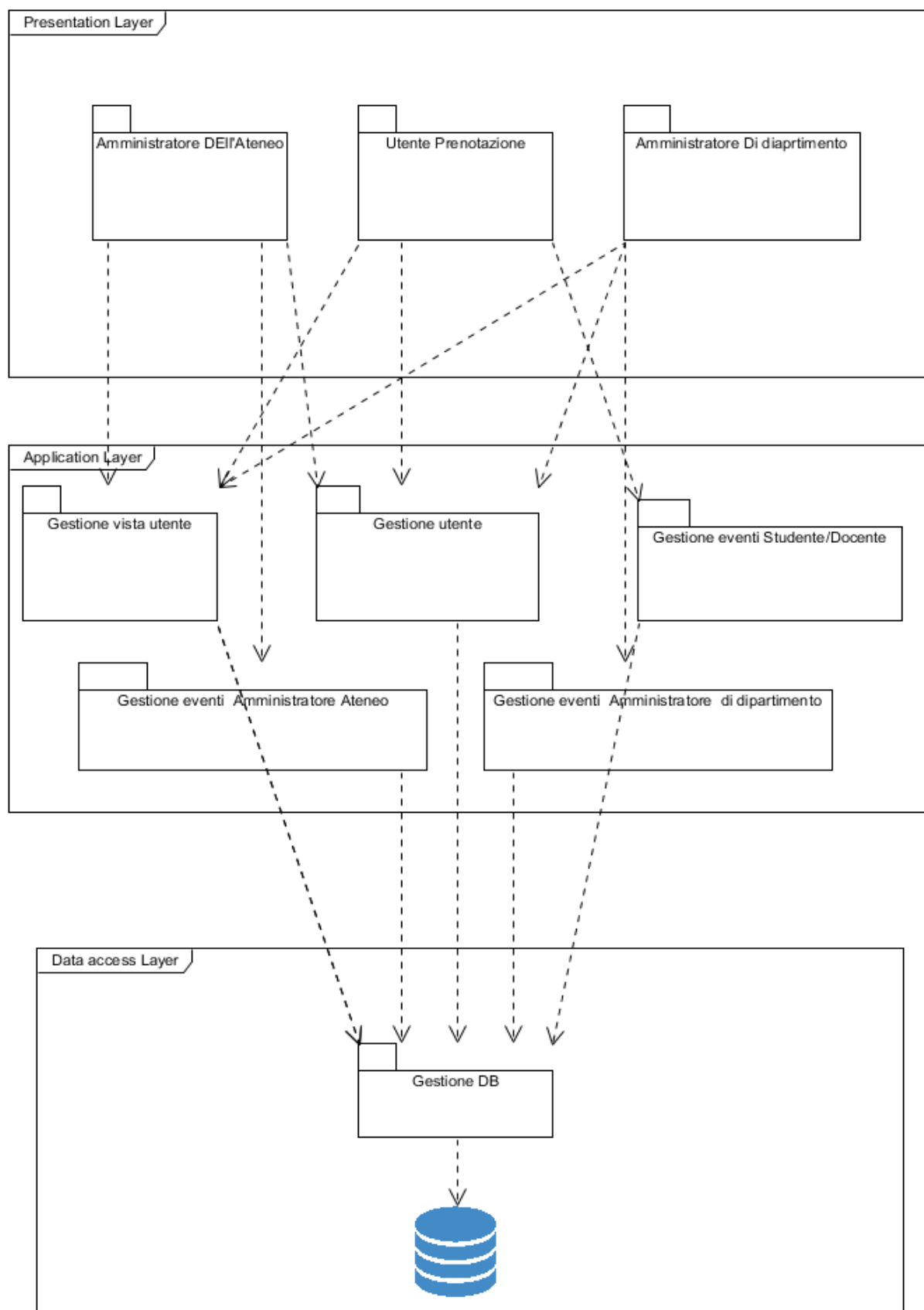
Presentiamo una decomposizione del nostro sistema in 3 layer (o livelli: Presentation Layer, Application Logic Layer e Data Access Layer che si occupano di gestire funzionalità ed aspetti differenti del nostro sistema:

- **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 e permette di visualizzare gli orari delle aule libere, il proprio profilo personale e l'interfaccia riguardante la richiesta di prenotazione di un'aula.
- **Amministratore dipartimento:** Gui che gestisce l'interfaccia dedicata al responsabile delle prenotazioni del dipartimento assegnato e permette di aggiungere o eliminare una prenotazione eseguita da un utente.
- **Amministratore Ateneo:** Gui che gestisce l'interfaccia dedicata al responsabile dell'ateneo e permette la creazione o eliminazione di aule, edifici e dipartimenti. Inoltre, può assegnare le funzionalità di amministratore ad un docente e di conseguenza assegnare un amministratore ad un dipartimento.
- **Application 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 il login e il logout, accedere ai servizi relativi all'area personale e la visualizzazione di quest'ultima.



- **Gestioni Eventi Amministratore:** Sottosistema che permette la gestione di tutte le funzionalità di un amministratore di dipartimento, quali accettazione o rifiuto di richieste di prenotazione, prenotazione di un'aula ed eliminazione di una prenotazione già accettata.
- **Gestione vista utenti:** Questo sottosistema permette la visualizzazione degli edifici, aule, storico prenotazioni, calendario e dipartimento.
- **Gestione eventi Docente/Studente:** Sottosistema che permette la richiesta di prenotazione di un'aula.
- **Gestione eventi Amministratore Ateneo:** Sottosistema che permette la gestione delle funzionalità dell'amministratore generale dell'ateneo 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 necessari al funzionamento dell'intero sistema ovvero i dati persistenti.
- **Gestione DataBase(DB):** sottosistema che si occupa di immagazzinare e di prelevare i dati persistenti dal nostro database.





**NB:** Nel diagramma l'utente non registrato fa parte del package utente prenotazione avendo solo la possibilità di visualizzazione del package gestione vista utente.



### 3.3 Mapping hardware/software

Per il progetto software che si vuole realizzare faremo uso di componenti "off-the-shelf", 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.

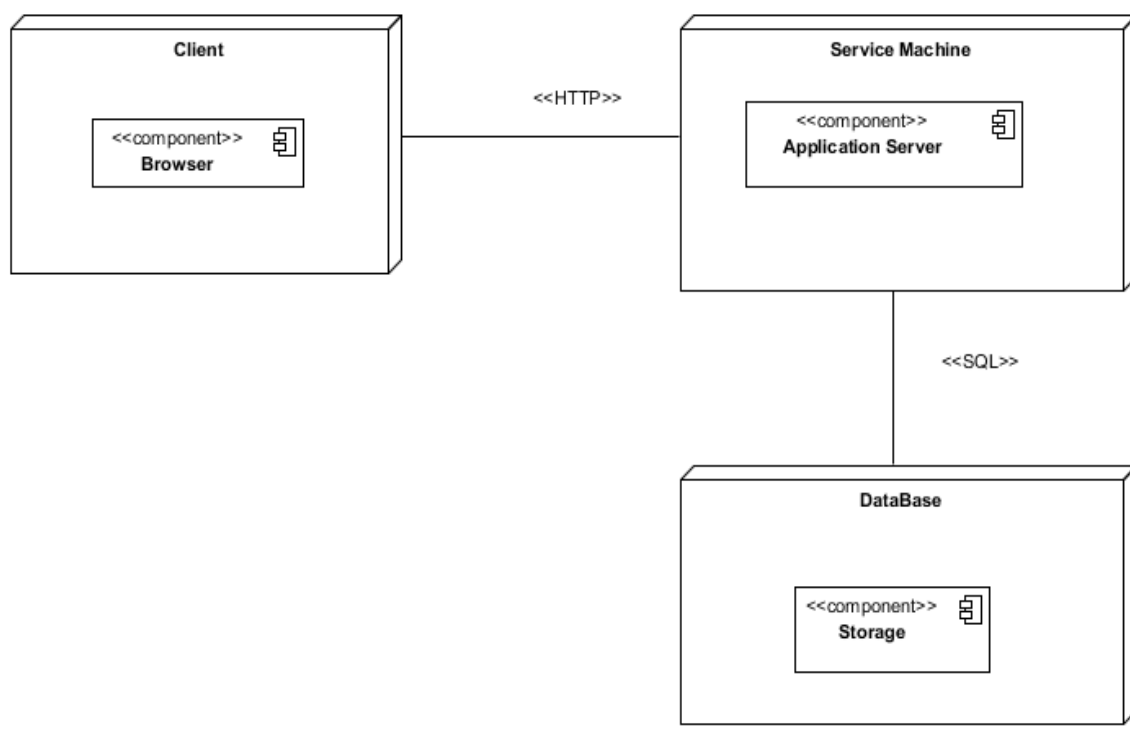
Javascript è un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client.

Per il sistema, basato su un'architettura MVC, sono state 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.

### 3.3.1 Deployment

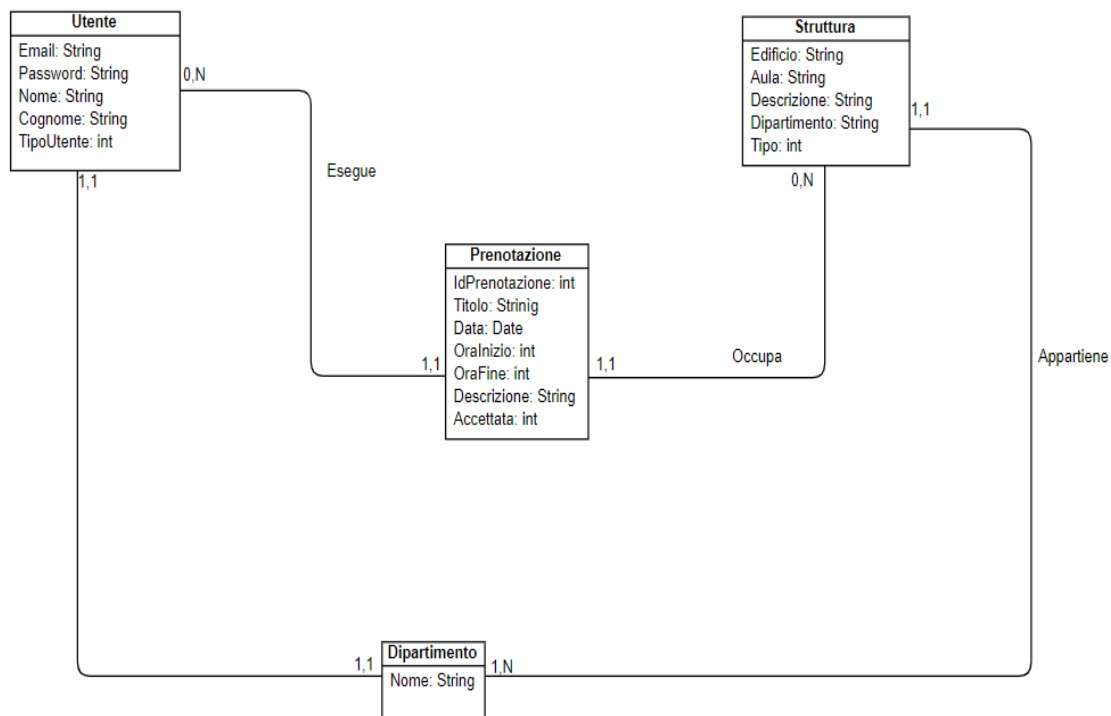
Gli utenti potranno interagire con Gestione prenotazioni tramite un WebBrowser collegandosi all'indirizzo del nostro sito. Il Control si occupa di: interpretare gli eventi generati dall'utente, richiedere e prelevare le opportune risorse dalla Base di Dati ed inviare le risposte tramite protocollo HTTP.



### 3.4 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

Attributo	Tipo	Vincoli
Email	String	Primary key
Password	String	NOT NULL
Nome	String	NOT NULL
Cognome	String	NOT NULL
TipoUtente*	Integer	NOT NULL

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

Tabella Prenotazione

Attributo	Tipo	Vincoli
IDprenotazione	Integer	PRIMARY KEY
Titolo	String	NOT NULL
Data	date	NOT NULL
Oralnizio	Integer	NOT NULL
Ora fine	Integer	NOT NULL
Descrizione	String	NOT NULL
NomeUtente	String	FOREIGN KEY UTENTE
Aula	String	FOREIGN KEY STRUTTURA
Edificio	String	FOREIGN KEY STRUTTURA
Accettata	Int	NOT NULL



Tabella Struttura

Attributi	Tipo	Vincoli
Edificio	String	PRIMARY KEY(composite key)
Aula	String	PRIMARY KEY(composite key)
Descrizione	String	
Dipartimento	String	FOREIGN KEY DIPARTIMENTO(Nome)
Tipo	Int	NOT NULL

Tabella Dipartimento

Attributi	Tipo	Vincoli
Nome	String	PRIMARY KEY
AmmDip	String	FOREIGN KEY UTENTE(Email)

### 3.5 Controllo degli accessi e sicurezza

Il sistema può essere utilizzato da qualsiasi utente possidente una connessione internet per il compimento di un insieme ristretto di funzionalità, che va ad allargarsi quando l'utente si registra alla piattaforma. La piattaforma predispone dei controlli per l'iscrizione al sistema, dando ad ogni utente la possibilità di registrarsi utilizzando una coppia di credenziali (e-mail e password, dove la mail deve necessariamente seguire dei pattern di nomenclatura cosicché il sistema possa verificare l'appartenenza dell'utente all'ateneo). L'accesso alla piattaforma per gli utenti è garantito dall'utilizzo di queste credenziali, e verrà chiesto ad 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.

Tutte le funzionalità del sistema sono descritte nella matrice degli accessi sottostante.

Attori	Gestione utente	Gestione eventi amministratore di dipartimento	Gestione vista degli utenti	Gestione eventi docente/studente	Gestione eventi amministratore ateneo
Utente non registrato	<<Registrazione>>		<<Visualizzazione edificio>>  <<Visualizzazione aula>>  <<Visualizzazione calendario>>  <<Visualizzazione dipartimento>>		
Utente (docente)	<<Autenticazione>>  <<Visualizzazione area personale>		<<Visualizzazione storico prenotazioni>>  <<Visualizzazione	<<Richiesta di prenotazione e aula>>	

	> <<Modifica password>>  <<Logout>>		edificio>> <<Visualizzazione aula>>  <<Visualizzazione calendario>> >  <<Visualizzazione dipartimento>>		
Utente (studente)	<<Autenticazione>>  <<Visualizzazione area personale>> >  <<Modifica password>>  <<Logout>>		<<Visualizzazione storico prenotazioni>>  <<Visualizzazione edificio>>  <<Visualizzazione aula>>  <<Visualizzazione calendario>> >  <<Visualizzazione dipartimento>>	<<Richiesta di prenotazione e aula>>	
Amministratore	<<Autenticazione>>  <<Visualizzazione area personale>> >  <<Modifica password>>  <<Logout>>	<<Accettazione richiesta di prenotazioni>>  <<Rifiuto richiesta di prenotazioni>>	<<Visualizzazione storico prenotazioni>>  <<Visualizzazione edificio>>  <<Visualizzazione aula>>		



		<<Visualizza zione richieste prenotazioni>>  <<Prenotazione aula>>  <<Eliminazione prenotazioni>>	<<Visualizzazione calendario>>  <<Visualizzazione dipartimento>>		
Amministratore Ateneo	<<Autenticazione>>  <<Visualizzazione area personale>>  <<Modifica password>>  <<Logout>>	<<Aggiungi aula lista studenti>>  <<Rimuovi aula lista studenti>>	<<Visualizzazione edificio>>  <<Visualizzazione aula>>  <<Visualizzazione calendario>>  <<Visualizzazione dipartimento>>		<<Aggiunta dipartimento>>  <<Rimozione dipartimento>>  <<Aggiunta edificio>>  <<Rimozione edificio>>  <<Aggiunta aula>>  <<Rimozione aula>>  <<Aggiunta aula a dipartimento>>  <<Rimozione aula a dipartimento>>  <<Visualizzazione elenco utenti>>

					<<Assegnazione titolo amministratore>>  <<Rimozione e titolo amministratore>>  <<Assegnazione amministrazione a dipartimento>>  <<Rimozione e amministrazione a dipartimento>>
--	--	--	--	--	--

### 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 di 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

#### SERVER:

- **Fase di Startup:**  
La fase di avvio del server Apache Tomcat dura circa trenta secondi, dopodiché viene avviato il server dove è contenuto il DBMS MySQL. La connessione al DBMS verrà effettuata tramite il driver JDBC. Successivamente viene avviata la JVM sul server e vengono letti i dati di configurazione del database. Infine, vengono caricati tutti i dati in memoria volatile e il sistema è pronto all'uso.
- **Fase di Terminazione:**  
Il server può essere spento in caso di guasto o in caso di necessità di manutenzione. Terminata l'attività di manutenzione o di riparazione del guasto il processo server viene avviato nuovamente.
- **Fallimento:**  
Il server può fallire in caso di interruzione inaspettata dell'alimentazione e/o di guasti hardware. Il server può andare in crash quando viene sollevata un'eccezione che non è stata gestita.

#### CLIENT:

- **Scenario "Startup":**  
Andrea è un tecnico che intende avviare la piattaforma Gestione Prenotazioni.  
Una volta avviato il web server Apache Tomcat, configura il DBMS MySQL per essere raggiunto dal medesimo web server. A questo punto il web server carica su Tomcat l'eseguibile della piattaforma Gestione prenotazioni. Dopo trenta secondi dall'avvio la piattaforma è operativa e raggiungibile dagli utenti.
- **Fase di Terminazione:**  
All'utente basterà chiudere la pagina del sistema per interrompere la comunicazione, interrompendo così la sessione TCP.
- **Scenario "Fallimento":**  
Il server mostra al client una schermata di errore. Per evitare la perdita di dati, viene comunicato al DBMS di effettuare una forzata scrittura su memoria fisica delle informazioni ancora presenti in memoria volatile. Il DBMS mantiene dei registri di log dove sono annotate tutte le operazioni sul DB.

## 4. Servizi dei Sottosistemi

### Gestione Utente

Sottosistema	Gestione Utente
Descrizione	Questo sottosistema permette di effettuare tutte le operazioni relative agli utenti registrati alla piattaforma.
Servizi offerti	
Servizio	Descrizione
Registrazione	Questo servizio consente la registrazione di qualsiasi utente che non sia già registrato.
Autenticazione	Questo servizio consente l'autenticazione di qualsiasi utente che sia già registrato.
Logout	Questo servizio consente il logout di qualsiasi utente che sia già registrato
Visualizzazione area personale	Questo servizio consente la visualizzazione dell'area personale di utente che sia già registrato
Modifica password	Questo servizio consente la modifica della password di un utente registrato.
Elimina Prenotazione	Questo servizio consente l'eliminazione diretta di una prenotazione già accettata dall'amministratore di dipartimento.



### Gestione eventi Amministratore di Dipartimento

Sottosistema	Gestione eventi Amministratore dipartimento
Descrizione	Questo sottosistema permette di effettuare tutte le operazioni relative agli utenti che hanno permessi di amministratore alla piattaforma.
Servizi offerti	
Servizio	Descrizione
Accetta richiesta di prenotazione	Questo servizio permette di accettare una richiesta di prenotazione.
Rifiuta richiesta di prenotazione	Questo servizio permette di rifiutare una richiesta di prenotazione.
Visualizza richieste di prenotazione	Questo servizio permette di visualizzare le richieste di prenotazione.
Aggiungi aule lista studenti	Questo servizio rende prenotabile un'aula anche agli studenti.
Rimuovi aule lista studenti	Questo servizio non rende prenotabile un'aula agli studenti.
Prenotazione aula	Questo servizio permette la prenotazione immediata di un'aula .



### Gestione vista degli utenti

Sottosistema	Gestione vista utenti
Descrizione	Questo sottosistema permette di effettuare tutte le operazioni di visualizzazione delle pagine del sito da parte degli utenti registrati e non registrati.
Servizi offerti	
Servizio	Descrizione
Visualizza storico prenotazioni	Questo servizio consente la visualizzazione dello storico prenotazioni di uno studente.
Visualizzazione edificio	Questo servizio consente la visualizzazione di un edificio dell'ateneo.
Visualizzazione aula	Questo servizio consente la visualizzazione di un'aula dell'ateneo.
Visualizzazione Calendario	Questo servizio consente la visualizzazione del calendario di un'aula.
Visualizzazione dipartimento	Questo servizio consente la visualizzazione di un dipartimento.



### Gestione eventi Docente/Studente

Sottosistema	Gestione eventi Docente/Studente
Descrizione	Questo sottosistema permette di effettuare le operazioni relative ad una richiesta di prenotazione.
Servizi offerti	
Servizio	Descrizione
Richiesta di prenotazione aula	Questo servizio consente la richiesta di prenotazione delle aule da parte di un Docente o uno studente.
Visualizzazione edificio	Questo servizio consente la visualizzazione di un edificio dell'ateneo.
Visualizzazione aula	Questo servizio consente la visualizzazione di un'aula dell'ateneo.
Visualizzazione Calendario	Questo servizio consente la visualizzazione del calendario di un'aula.
Visualizzazione dipartimento	Questo servizio consente la visualizzazione di un dipartimento.

## Gestione eventi Amministratore Ateneo

Sottosistema	Gestione eventi Amministratore Ateneo
Descrizione	Questo sottosistema permette di effettuare le operazioni relative ad una richiesta di prenotazione.
Servizi offerti	
Servizio	Descrizione
Aggiunta Dipartimento	Questo servizio consente l'aggiunta di un nuovo dipartimento nella piattaforma.
Rimozione Dipartimento	Questo servizio consente la rimozione di un dipartimento nella piattaforma.
Aggiunta Edificio	Questo servizio consente l'aggiunta di un nuovo edificio nella piattaforma.
Rimozione edificio	Questo servizio consente la rimozione di un edificio nella piattaforma.
Aggiunta aula	Questo servizio consente l'aggiunta di una nuova aula nella piattaforma.
Rimozione aula	Questo servizio consente la rimozione di un'aula nella piattaforma.
Aggiunta aula dipartimento	Questo servizio consente di associare delle aule ad un dipartimento.
Rimozione aula dipartimento	Questo servizio consente di dissociare delle aule da un dipartimento.
Visualizzazione elenco utenti	Questo servizio consente la visualizzazione di tutti gli utenti iscritti alla piattaforma.
Assegnazione elenco utenti	Questo servizio consente la visualizzazione di tutti gli utenti iscritti alla piattaforma.
Assegnazione titolo amministratore	Questo servizio consente l'assegnazione del titolo di amministratore di dipartimento ad un utente di tipo docente registrato sulla piattaforma.
Rimozione titolo amministratore	Questo servizio consente la rimozione del titolo di amministratore di dipartimento ad un utente di tipo docente registrato sulla piattaforma.





Assegnazione amministratore dipartimento	Questo servizio consente di assegnare un amministratore ad un dipartimento già esistente.
Rimozione amministratore dipartimento	Questo servizio consente di dissociare un amministratore da un dipartimento già esistente.



## 5. Glossario

**Amministratore:** Utilizza il sistema per l'accettazione o per il rifiuto di una prenotazione.

**Amministratore Ateneo:** Utilizza il sistema per la gestione e per l'inserimento, rimozione o modifica di amministratori, edifici, dipartimenti ed aule.

**Application Logic Layer:** si occupa della gestione della logica del Sistema.

**Aula libera:** Viene indicata un'aula dell'ateneo che non presenta avere lezioni, tutorati convegni o altro, quindi libera di poter essere prenotata.

**Client:** componente che accede a servizi e risorse di un altro componente detto server.

**Data access layer:** si occupa della gestione dei dati persistenti e dello scambio di dati tra i sottosistemi.

**DBMS:** sistema software per la creazione, manipolazione e interrogazione efficiente di database.

**HTTP:** protocollo di trasferimento di ipertesti che consente a due macchine, client e server, di interagire attraverso un meccanismo di richiesta/risposta. Il client inoltra una richiesta al server, che verrà soddisfatta con la risposta di quest'ultimo.

**JSP:** tecnologia di programmazione web utilizzata per fornire contenuti dinamici.

**Layer:** suddivisione del sistema in livelli (o strati).

**Piattaforma:** Definisce l'insieme delle funzionalità fornite dal sistema attraverso l'applicazione web.

**Presentation layer:** si occupa della visualizzazione della piattaforma.

**RAD (Requirement Analysis Document):** documento contenente informazioni inerenti al sistema da realizzare raccolte durante la fase di Requirement Analysis e Requirement Elicitation.

**SDD (System Design Document):** Documento formalizzato alla definizione di obiettivi di progettazione del sistema, decomposizione del sistema in sottosistemi più piccoli e scelta di architettura software più adatta al sistema.

**Server:** componente che gestisce traffico di informazioni e fornisce servizi e risorse attraverso la rete.

**Servlet:** oggetti Java all'interno del server web che permettono di creare web applications in combinazione con JSP.

**Sistema:** Definisce la totalità delle componenti dal punto di vista strutturale/implementativo.

**Utente:** Un utilizzatore del sistema che sia esso un amministratore, un docente o uno studente.

**Visitatore:** Utente non registrato visualizza informazioni.

**Web Browser:** applicazione software installata sul client che permette di visualizzare e navigare le risorse del web.