* 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. **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 portabili, 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 |  |
| 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 Costi di sviluppo | Costo totale del sistema. **(?)** | Costo |  |
| DG\_10 Estendibilità | Si potranno inserire nuove funzionalità al sistema. | Supportabilità | NFR\_13 |
| DG\_11 Tracciabilità dei requisiti | Uso di una matrice di tracciabilità. **(?)** | Manutenzione |  |
| DG\_12 Affidabilità | Il sistema garantirà una corretta gestione delle funzionalità. | Affidabilità | NFR\_8 |
| DG\_13 Manutenibilità | Capacità del sistema di essere facilmente ripristinato in caso sia necessario realizzare un intervento di manutenzione. | Supportabilità | NFR\_13 |
| DG\_14 Adattabilità | Il sistema dovrà adattarsi facilmente alle nuove necessità del cliente. | Implementazione | NFR\_18 |

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

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

Al secondo punto 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.

Attualmente l’unica piattaforma che esiste permette solo la visualizzazione delle aule, il sistema 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 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 delle aule dell’ateneo direttamente 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.



**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 e 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.

●**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 e 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, l’eliminazione di una prenotazione già accettata.

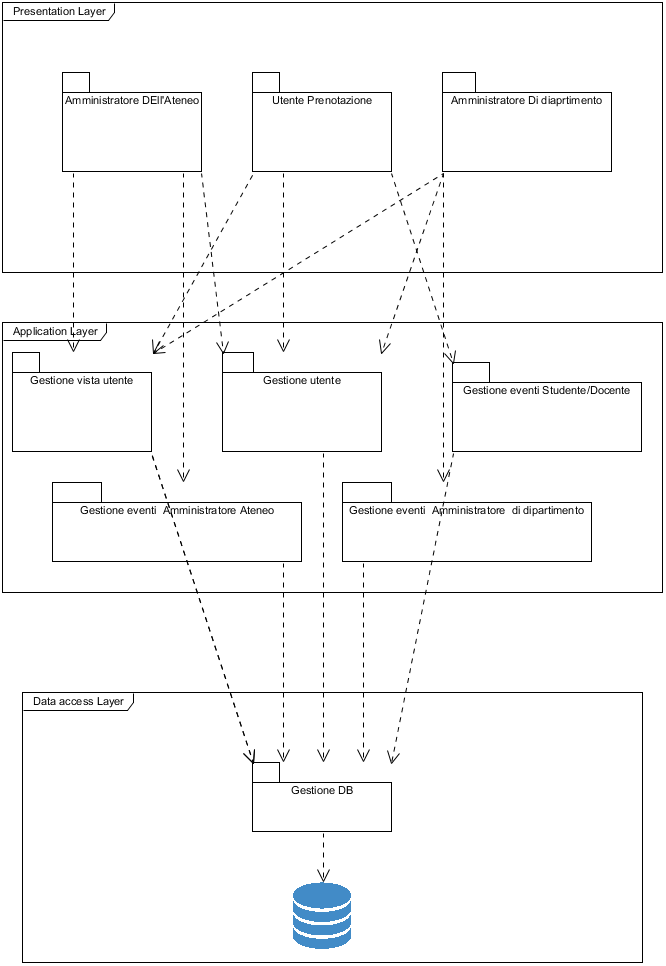
**Gestione vista degli utenti:**Questo sottosistema permette la visualizzazione degli edifici, aule dello storico prenotazioni per utente, del calendario e del dipartimento.

**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 e l’assegnazione degli amministratori ai dipartimenti e la suddetta carica di amministratore ai docenti.

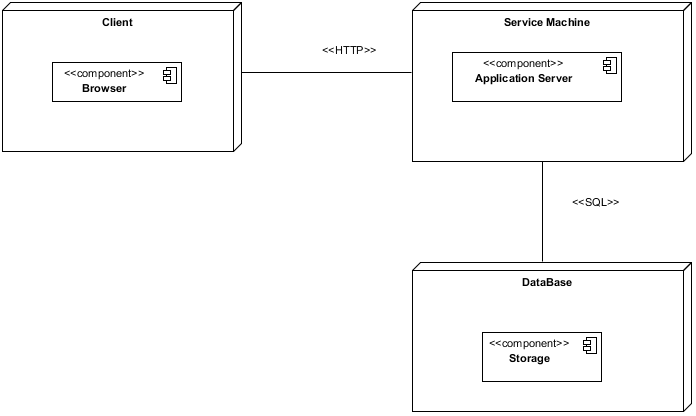
● **Data Access Layer(Model) :** Livello che gestisce i dati che sono necessari al funzionamento dell’intero sistema, ovvero i dati persistenti.

**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 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.

**3.4 Gestione dati persistenti**

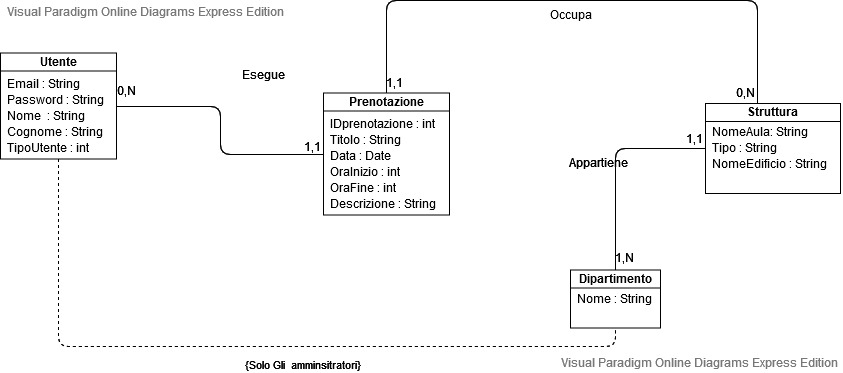
Il sistema che si vuole sviluppare sarà installato su un solo computer e utilizzerà un

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 |
| OraInizio | Integer | NOT NULL |
| Ora fine | Integer | NOT NULL |
| Descrizione | String | NOT NULL |

**Tabella Struttura**

|  |  |  |
| --- | --- | --- |
| Attributi | Tipo | Vincoli |
| NomeAula | String | PRIMARY KEY |
| NomeEdificio | String | UNIQUE NOT NULL |
| TipoAula | String | NOT NULL |

**Tabella Dipartimento**

|  |  |  |
| --- | --- | --- |
| Attributi | Tipo | Vincoli |
| Nome | String | PRIMARY KEY |

**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 generale** |
| Utente non registrato | -Registrazione |  | -Visualizzazione edificio  -Visualizzazione aula  -Visualizzazione calendario  -Visualizzazione dipartimento |  |  |
| Utente (docente) | -Autenticazione  -Visualizzazione area personale  -Modifica password  -Logout |  | -Visualizza storico prenotazioni  -Visualizzazione edificio  -Visualizzazione aula  -Visualizzazione calendario  -Visualizzazione dipartimento | -Richiesta di prenotazione aula |  |
| Utente (studente) | -Autenticazione  -Visualizzazione area personale  -Modifica password  -Logout |  | -Visualizza storico prenotazioni  -Visualizzazione edificio  -Visualizzazione aula  -Visualizzazione calendario  -Visualizzazione dipartimento | -Richiesta di prenotazione aula |  |
| Amministratore | -Autenticazione  -Visualizzazione area personale  -Modifica password  -Logout | -Accettazione richiesta di prenotazione  -Rifiuto richiesta di prenotazione  -Visualizzazione richieste prenotazione  -Prenotazione aula  -Eliminazione prenotazione | -Visualizza storico prenotazioni  -Visualizzazione edificio  -Visualizzazione aula  -Visualizzazione calendario  -Visualizzazione dipartimento |  |  |
| Amministratore ateneo | -Autenticazione  -Visualizzazione area personale  -Modifica password  -Logout | -Aggiungi aule lista studenti  -Rimuovi aule lista studenti | -Visualizzazione edificio  -Visualizzazione aula  -Visualizzazione calendario  -Visualizzazione dipartimento |  | -Aggiunta dipartimento  -Rimozione dipartimento  -Aggiunta edificio  -Rimozione edificio  -Aggiunta aula  -Rimozione aula  -Aggiunta aule a dipartimento  -Rimozione aule a dipartimento  -Visualizzazione elenco utenti  -Assegnazione titolo amministratore  -Rimozione titolo amministratore  -Assegnazione amministrazione a dipartimento  -Rimozione 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**

3.7 Condizioni limite

SERVER

Fase di Startup:

La fase di avvio del server Apache Tomcat dura circa trenta secondi, dopo di che viene avviato il server dove è contenuto il DBMS MySQL. Viene effettuata la connessione al DBMS 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. 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

Fase di Terminazione:

All’utente basterà chiudere la pagina del sistema per interrompere la comunicazione, interrompendo così la sessione TCP.

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, carica su Tomcat l’eseguibile della piattaforma Gestione prenotazioni. Dopo trenta secondi dall’avvio la piattaforma è operativa e raggiungibile dagli utenti.

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**

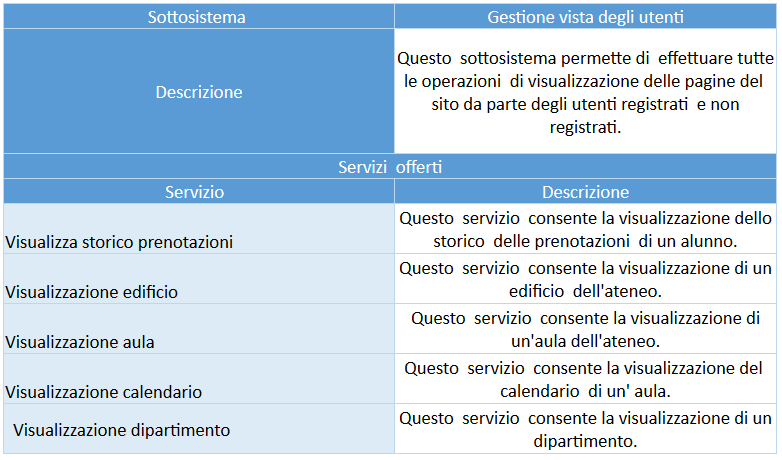
Gestione Utente

|  |  |
| --- | --- |
| **Sottosistema** | **Gestione Utente** |
| **Descrizione** | Questo sottosistema permette di effettuare tutte le operazioni relative agli utenti registrati alla piattaforma. |
| **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. |

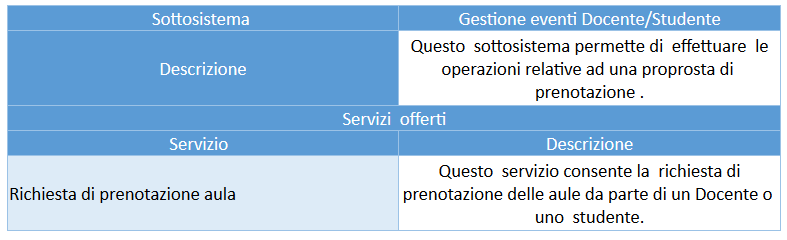
Gestione eventi Amministratore di Dipartimento

|  |  |
| --- | --- |
| **Sottosistema** | **Gestione eventi Amminstratore di dipartimento** |
| **Descrizione** | Questo sottosistema permette di effettuare tutte le operazioni relative agli utenti che hanno permessi di amministratore alla piattaforma. |
| **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 di un aula. |

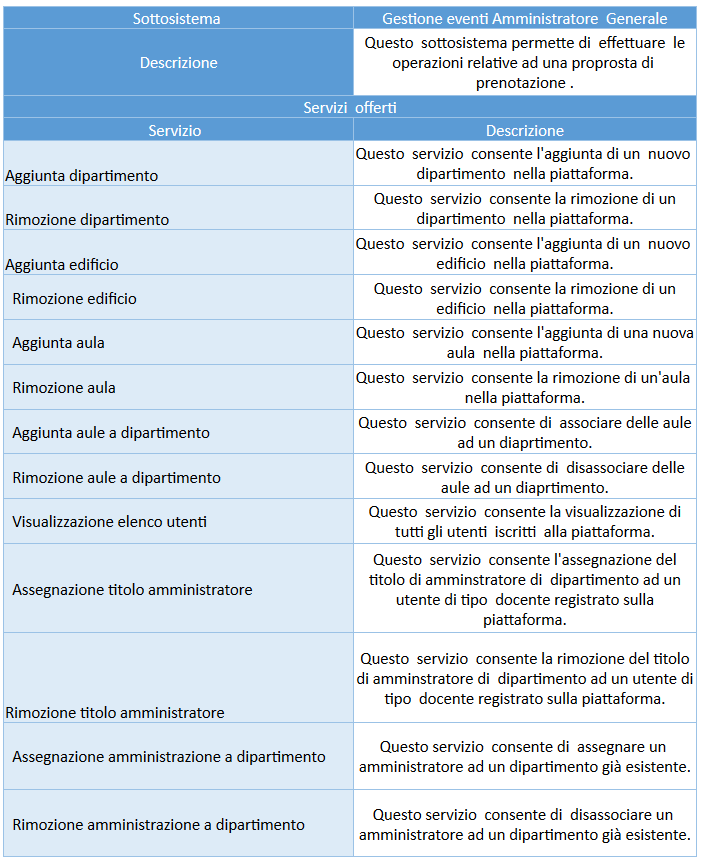
Gestione vista degli utenti



Gestione eventi Docente/Studente



Gestione eventi Amministratore dell’Ateneo



**5. Glossario**

**Utente aula** : Tutti gli studenti o docenti che fanno parte di dell’ateneo e che sono registrate alla piattaforma di prenotazione.

**Amministratore di dipartimento**: Soggetto che si occupa di gestire le segnalazioni effettuate dagli utenti.

**Amminsitratore dell’ateneo:** Soggetto che si occupa di gestire gli edifici e gli amministratori di diaprtimento dell’ateneo.