

Università degli Studi di Salerno

Corso di Ingegneria del Software

UniMeet
System Design
Versione 1.0



GitHub del progetto: [/UniMeet](#)

Progetto UniMeet Ingegneria del Software Prof A. De Lucia

Data: 20/11/2024

Progetto: UniMeet	Versione: 1.0
Documento: System Design	Data: 20/11/2024

Partecipanti:

Nome	Matricola
Lorenza Rosa Pia Natale	0512116647
Giovanni Tufano	0512112027
Antonio Del Vecchio	0512118501
Ciro Danzilli	0512111007

Scritto da:	Tutti i membri del gruppo
--------------------	---------------------------

Revision History

Data	Versione	Descrizione	Autore
14/10/2024	1.0	Problem Statement	Tutto il gruppo
28/10/2024	1.0	Requisiti e casi d'uso	Tutto il gruppo
11/11/2024	1.0	Requirements Analysis Document (RAD)	Tutto il gruppo
19/11/2024	1.1	Revisione RAD	Tutto il gruppo
20/11/2024	1.0	System Design	Tutto il gruppo

1. Introduzione	4
1.1 Scopo del software.....	4
1.2 Obiettivi del sistema.....	4
1.2.1 Criteri di Performance	4
1.2.2 Criteri di Affidabilità.....	4
1.2.3 Criteri di Manutenzione	5
1.3 Referenze	5
1.4 Panoramica	5
2. Architettura Software attuale.....	5
3. Architettura del software proposta	5
3.1 Overview	5
3.2 Subsystem decomposition	6
3.3 Hardware/software mapping.....	7
3.4 Persistent data management	7
3.5 Access control and security	9
3.5.1 Gestione Autenticazione	9
3.5.2 Gestione Prenotazione	10
3.6 Global software control.....	10
3.7 Boundary conditions.....	11
3.7.1 Servizi dei sottosistemi.....	11
4. Subsystem services glossary.....	12

1. Introduzione

1.1 Scopo del software

L'organizzazione dei ricevimenti tra docenti e studenti rappresenta una sfida logistica per molte università. Gli studenti spesso si trovano in difficoltà nel prenotare un appuntamento a causa dell'assenza di una piattaforma centralizzata, di tempistiche rigide o di problemi di comunicazione con i docenti. Al contempo, i docenti necessitano di strumenti che consentano loro di gestire la disponibilità in modo efficiente, ricevendo notifiche tempestive per evitare sovrapposizioni, appuntamenti mancati o disguidi.

UniMeet nasce per semplificare e ottimizzare questo processo. Il software offre una piattaforma intuitiva che consente a docenti e studenti di gestire prenotazioni e appuntamenti in modo dinamico e organizzato, riducendo errori, sovrapposizioni e problematiche di comunicazione.

1.2 Obiettivi del sistema

1.2.1 Criteri di Performance

- Tempo di risposta: Il sistema deve essere reattivo e in grado di servire più utenti contemporaneamente.
- Usabilità: Il sistema deve essere accessibile a quanti più utenti possibile, in modo da poter permettere a chiunque di poter inserire o visionare prodotti in maniera semplice e guidata da qualunque dispositivo utilizzato.

1.2.2 Criteri di Affidabilità

- Sicurezza: Il sistema deve garantire la protezione dei dati sensibili scambiati con l'utente ed evitare che terzi vi accedano, attraverso anche paradigmi di programmazione crittografati.
- Disponibilità: Il sistema deve essere disponibile ad un alto numero di utenti.
- Robustezza: Il sistema deve essere in grado di gestire correttamente l'immissione di eventuali input errati.

1.2.3 Criteri di Manutenzione

- Modificabilità: Il sistema deve poter essere facilmente modificabile in modo da correggere eventuali errori.
- Resistenza agli errori: Il sistema deve gestire in modo corretto eventuali eccezioni software.

1.3 Referenze

Riferimento al Requirement Analysis Document di UniMeet.

1.4 Panoramica

Il seguente documento di System Design (SDD) illustra i dettagli tecnici del design del sistema UniMeet. Ulteriori informazioni sulle funzionalità e le caratteristiche del sistema possono essere reperite nel documento di analisi dei requisiti (RAD), mentre una panoramica generale è fornita nel Problem Statement. In quest'ultimo documento viene presentata un'introduzione all'architettura e agli obiettivi di design che il sistema intende perseguire. Viene anche proposta una suddivisione del sistema in sottosistemi, con la definizione del mapping Hardware/Software, al fine di assegnare ciascun sottosistema a specifici componenti hardware. Inoltre, vengono descritti il controllo dell'accesso e le problematiche di sicurezza associate al sistema, con particolare attenzione al controllo generale del software e alla gestione dei Boundary, affrontando anche gli stati iniziali e la gestione complessiva del sistema.

2. Architettura Software attuale

Allo stato attuale ancora non è presente un'architettura software del progetto.

3. Architettura del software proposta

3.1 Overview

Per lo sviluppo del software utilizzeremo l'architettura **MVC (Model View Controller)** al fine di separare la logica di presentazione dalla logica di business del sistema. Il pattern MVC è largamente utilizzato nello sviluppo di applicazioni software per il web.

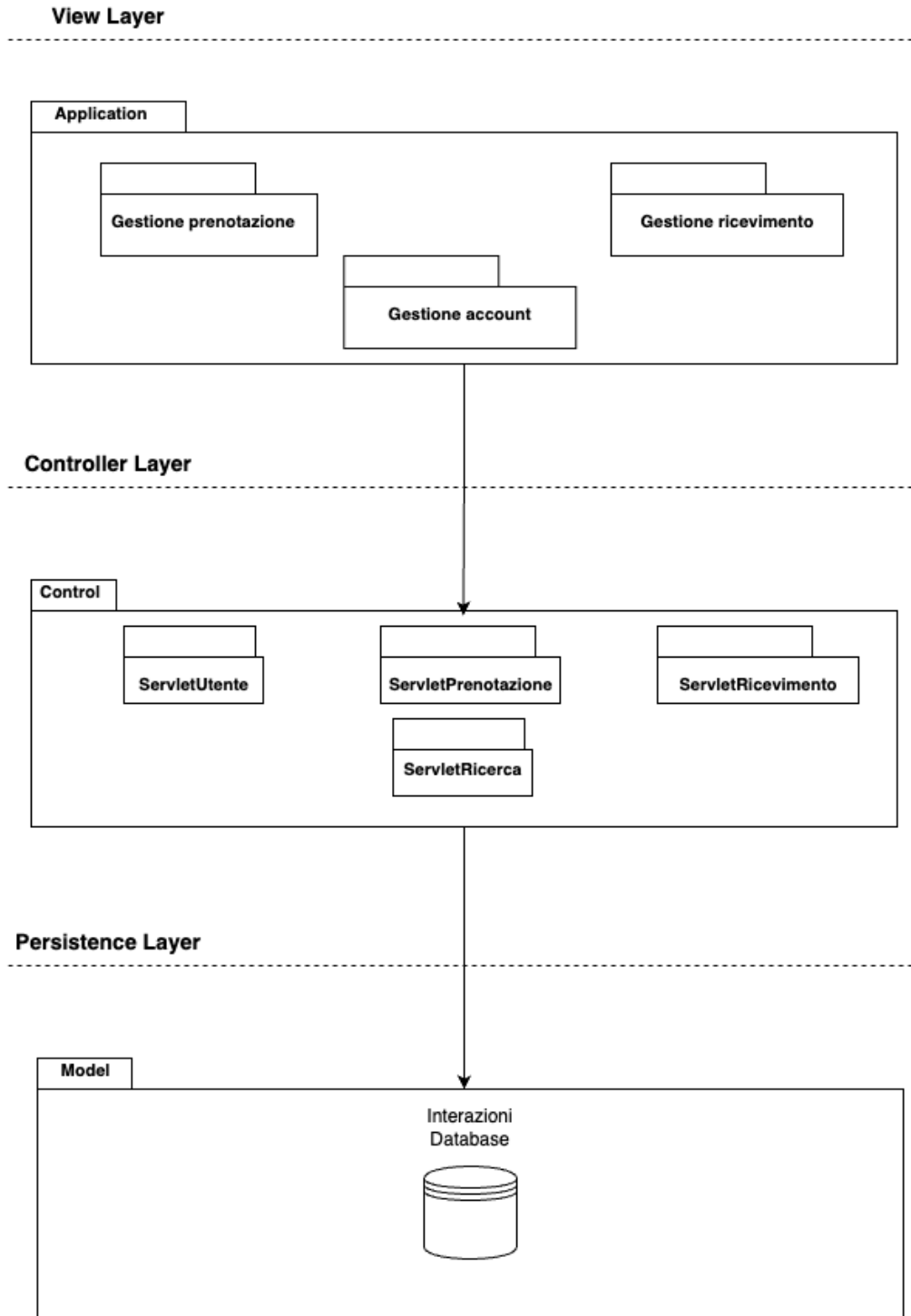
- **View:** Nella view gestiremo l'interazione con l'utente gestendo tutte le logiche relative all'interazione Utente-Sistema, nello specifico utilizzeremo delle pagine JSP miste a CSS per l'interfaccia utente.
- **Model:** In model gestiremo i dati relativi al sistema, manipoleremo il database mediante

Progetto UniMeet Ingegneria del Software Prof A. De Lucia

inserimento, modifica o cancellazione dei dati presenti in esso.

- **Controller:** Mediante il controller gestiremo tutte le interazioni che avvengono tra l'utente del sistema e il sistema, è il ponte diretto tra l'utente ed i dati sottostanti.

3.2 Subsystem decomposition

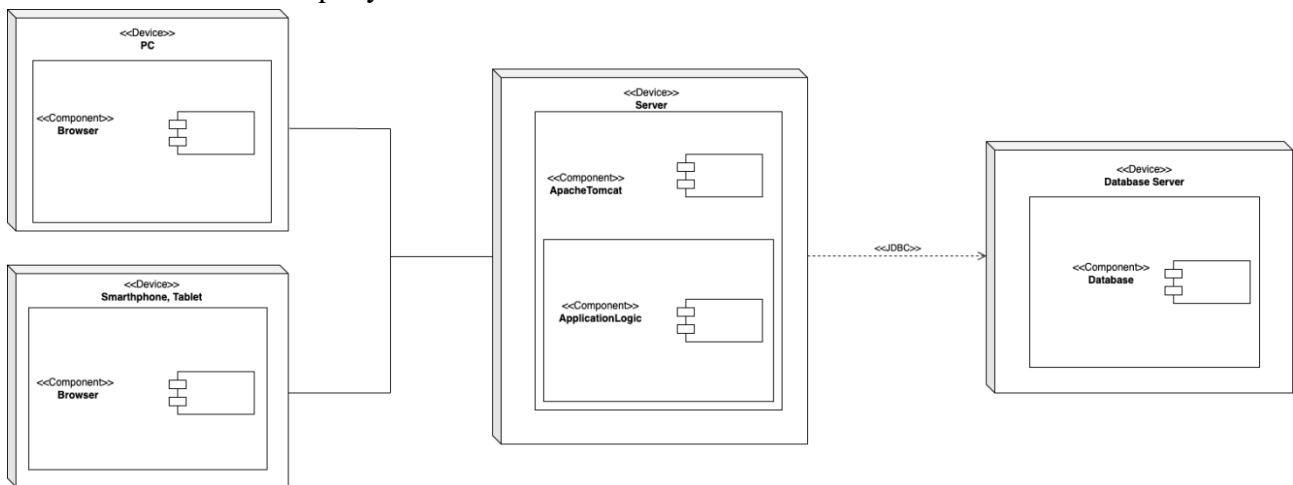


Gestione account	Aggrega tutte le funzioni relative a: Login, logout, registrazione, modifica password.
-------------------------	--

Gestione prenotazione	Aggrega tutte le funzioni relative a: Prenotazione ricevimento, conferma prenotazione e rifiuto prenotazione.
Gestione ricevimento	Funzioni relative all'aggiunta di un ricevimento

3.3 Hardware/software mapping

Il sistema utilizza un'architettura Client/Server. Il web Server è rappresentato da Apache Tomcat 9 ed è situato su una singola macchina, la logica del sistema è costituita da Java Servlet mentre l'interfaccia utente è realizzata usando JSP (Java Server Page), il Client è rappresentato da Web Browser usato dall'utente. La comunicazione tra i nodi è rappresentata da richieste e risposte http tra Client e Server e da query in JDBC tra server e database.



3.4 Persistent data management

3.4.1 Descrizione delle entità persistenti

1.1 Studente

- matricola INTEGER PRIMARY KEY
- nome VARCHAR(30)
- cognome VARCHAR(30)
- email VARCHAR(30)
- passwordHash VARCHAR(255)
- domandaSicurezza VARCHAR(30)
- risposta VARCHAR(30)

Rappresenta uno studente che accede a UniMeet

1.2 Professore

- codice INTEGER PRIMARY KEY
- nome VARCHAR(30)
- cognome VARCHAR(30)
- ufficio VARCHAR(30)
- email VARCHAR(30)

- password_hash VARCHAR(255)
- domandaSicurezza VARCHAR(30)
- risposta VARCHAR(30)

Rappresenta un Professore che accede a UniMeet

1.3 Insegnamento

- codice INTEGER PRIMARY KEY
- nome VARCHAR(30)
- codiceProfessore INTEGER
- FOREIGN KEY(codiceProfessore) REFERENCES professore(codice)

Rappresenta un insegnamento di un professore

1.4 Ricevimento

- codice INTEGER PRIMARY KEY ,
- giorno VARCHAR(30)
- ora VARCHAR(30)
- note VARCHAR(200)
- codiceProfessore INTEGER
- FOREIGN KEY(codiceProfessore) REFERENCES professore(codice)

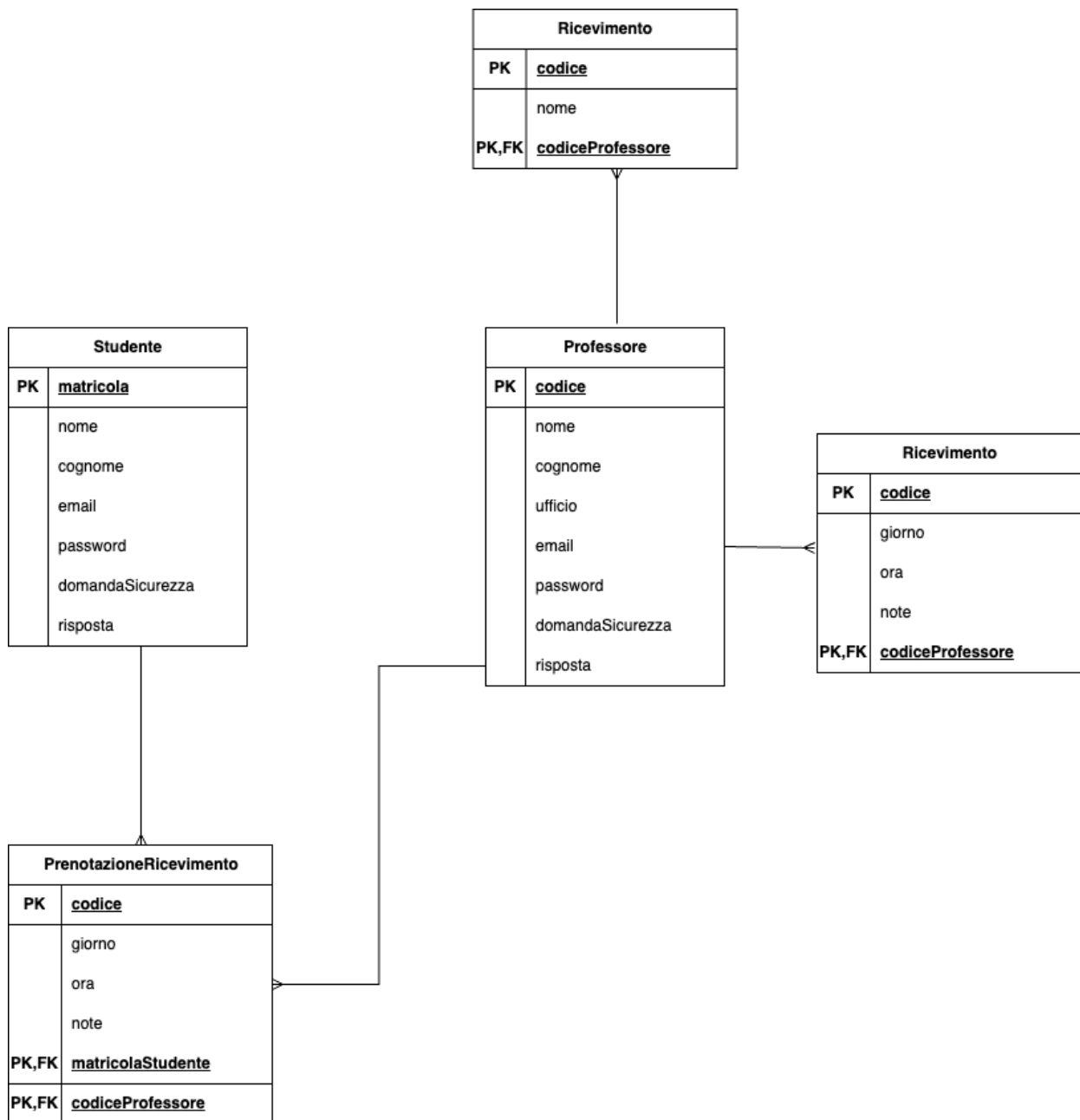
Rappresenta un ricevimento messo a disposizione da un professore

1.5 PrenotazioneRicevimento

- codice INTEGER PRIMARY KEY
- giorno VARCHAR(30)
- ora VARCHAR(30)
- note VARCHAR(200)
- codiceProfessore INTEGER
- matricolaStudente INTEGER
- FOREIGN KEY (codiceProfessore) REFERENCES professore(codice)
- FOREIGN KEY (matricolaStudente) REFERENCES studente(matricola)

Rappresenta la prenotazione effettuata da uno studente per un ricevimento

3.4.2 Schema Logico



3.5 Access control and security

3.5.1 Gestione Autenticazione

	Ospite	Studiante	Docente
--	--------	-----------	---------

Gestione Utente	<ul style="list-style-type: none"> • Registrazione • Visualizza ricevimenti 	<ul style="list-style-type: none"> • Registrazione • Login • Logout • Ricerca • Riepilogo prenotazioni 	<ul style="list-style-type: none"> • Registrazione • Login • Logout • Gestione ricevimenti
-----------------	---	---	--

L'ospite può navigare in maniera limitata, infatti può solo visualizzare i ricevimenti e registrarsi al sito.

Gli utenti registrati hanno delle funzionalità divise in base al tipo di utente: **studente e docente**.

Lo studente dopo aver effettuato la registrazione e di conseguenza il login, oltre a visualizzare e ricercare i ricevimenti disponibili, può prenotare un ricevimento, avere accesso alla propria pagina personale e avere un riepilogo delle prenotazioni.

Il docente dopo aver effettuato la registrazione e di conseguenza il login, può visualizzare e gestire i ricevimenti, visualizzare la sua pagina personale e quindi avere accesso al riepilogo dei suoi appuntamenti.

3.5.2 Gestione Prenotazione

	Ospite	Studente	Docente
Gestione Ricevimento	<ul style="list-style-type: none"> • Prenotazione con richiesta di login 	<ul style="list-style-type: none"> • Prenotazione • Cancellazione 	<ul style="list-style-type: none"> • Aggiunta Ricevimento • Modifica Ricevimento • Accetta/rifiuta Ricevimento
Gestione Ricerca	<ul style="list-style-type: none"> • Ricerca 	<ul style="list-style-type: none"> • Ricerca 	<ul style="list-style-type: none"> • Ricerca
Riepilogo Ricevimenti		<ul style="list-style-type: none"> • Riepilogo 	<ul style="list-style-type: none"> • Riepilogo • Accettazione/Rifiuto Ricevimento

L'accesso alla sezione ricevimenti è limitato: l'utente ospite può visualizzare i ricevimenti disponibili, mentre le altre funzionalità sono riservate agli utenti registrati, ossia studenti e docenti. La ricerca è invece disponibile per tutti gli utenti, indipendentemente dalla registrazione.

3.6 Global software control

UniMeet è una piattaforma web in cui il Web Server (Tomcat) riceve e gestisce le richieste degli utenti verso le JSP, indirizzandole alle Servlet competenti. Le Servlet elaborano la richiesta,

Progetto UniMeet Ingegneria del Software Prof A. De Lucia

interagiscono con il Model (DAO) quando necessario, e preparano la risposta. Una volta completata l'elaborazione, la Servlet esegue un redirect alla pagina JSP corrispondente, che viene visualizzata dall'utente con la risposta generata.

3.7 Boundary conditions

Inizializzazione del sistema: l'amministratore avvia il server con il comando di startup. In questo modo il sistema è pronto a fornire tutti i servizi necessari al fine di garantire il corretto funzionamento del sistema.

Spegnimento del sistema: l'amministratore si occupa di terminare l'avvio del server con il comando di spegnimento, dopodiché verrà reso inutilizzabile e chiude tutti i servizi.

3.7.1 Servizi dei sottosistemi

- Gestione account

Registrazione (studente)	<ul style="list-style-type: none">• Consente ad uno studente di registrarsi e prenotare i ricevimenti.
Registrazione (professore)	<ul style="list-style-type: none">• Consente ad un docente di registrarsi ed accettare prenotazione da parte degli studenti o aggiungerne di nuove.
Login	<ul style="list-style-type: none">• Permette all'utente di accedere al sistema e svolgere operazioni a lui consentite.
Logout	<ul style="list-style-type: none">• Consente all'utente di uscire dal sistema.

- Gestione prenotazione

Prenota ricevimento	<ul style="list-style-type: none">• Lo studente può prenotare un ricevimento col docente selezionato.
Ricerca	<ul style="list-style-type: none">• L'utente può cercare il relativo docente per visualizzare la disponibilità.
Riepilogo prenotazioni	<ul style="list-style-type: none">• L'utente può avere un resoconto delle varie prenotazioni.
Cancella prenotazioni (studente)	<ul style="list-style-type: none">• Lo studente può annullare la prenotazione
Aggiunta/rimozione ricevimento (docente)	<ul style="list-style-type: none">• Il docente può scegliere di aggiungere o rimuovere un orario per il proprio ricevimento in base alla sua disponibilità.

Accettazione/rifiuto ricevimento (docente)	<ul style="list-style-type: none"> • Il docente può accettare o rifiutare un ricevimento.
---	--

4. Subsystem services glossary

4.1 Gestione Autenticazione

- **Registrazione (*studente*)**
 - Consente ad uno studente di registrarsi e prenotare i ricevimenti.
- **Registrazione (*professore*)**
 - Consente ad un docente di registrarsi ed accettare prenotazione da parte degli studenti o aggiungerne di nuove.
- **Login**
 - Permette all'utente di accedere al sistema e svolgere operazioni a lui consentite.
- **Logout**
 - Consente all'utente di uscire dal sistema.

4.2 Gestione Prenotazione

- **Prenota Ricevimento (*studente*)**
 - Lo studente può prenotare un ricevimento col docente selezionato.
- **Ricerca**

Progetto UniMeet Ingegneria del Software Prof A. De Lucia

- o L'utente può cercare il relativo docente per visualizzare la disponibilità.
- **Riepilogo prenotazioni**
 - o L'utente può avere un resoconto delle varie prenotazioni.
- **Cancella prenotazioni (*studente*)**
 - o Lo studente può cancellare una propria prenotazione.
- **Aggiunta/rimozione ricevimento (*docente*)**
 - o Il docente può scegliere di aggiungere o rimuovere un orario per il proprio ricevimento in base alla sua disponibilità.
- **Accettazione/rifiuto ricevimento (*docente*)**
 - o Il docente può accettare o rifiutare un ricevimento.