



# SDD System Design Document

Safe Meeting

SDD 2.1 – ultima modifica 07/01/2019

Destinatario del documento: Top Manager – Prof.ssa F.Ferrucci

Presentato da:

Alfonso Serio	Emilio Mainardi	Gianluca Verlingieri	Luca Di Chiara
Alessandra Capo	Donato Marmora	Andrea Califano	Matteo Ferrara

Approvato da:

Roberta Gesumaria	Francesca Tassatone
-------------------	---------------------



## Storia dei cambiamenti

Data	Versione	Cambiamenti	Autore
26/11/2018	SDD 0.1	Strutturazione documento ed inserimento delle sezioni relative all'introduzione e all'architettura del sistema corrente.	Tutti
27/11/2018	SDD 0.2	Stesura ed inserimento nel documento delle sezioni: obiettivi del sistema, mapping hw/sw, panoramica, divisione in sottosistemi, decomposizione in layer, diagramma di deployment, architettura sistema corrente.	Emilio Mainardi, Alfonso Serio, Alessandra Capo, Donato Marmora, Andrea Califano, Matteo Ferrara
29/11/2018	SDD 0.3	Inserimento nel documento della sezione: gestione dei dati persistenti.	Emilio Mainardi, Gianluca Verlingieri, Luca Di Chiara
30/11/2018	SDD 0.4	Inserimento nel documento delle sezioni: controllo accesso e sicurezza, controllo flusso globale e condizioni limite, servizi dei sottosistemi.	Matteo Ferrara, Donato Marmora, Gianluca Verlingieri
01/12/2018	SDD 0.5	Inserimento nel documento: definizioni, abbreviazioni, acronimi e riferimenti	Alfonso Serio
02/12/2018	SDD 1.0	Revisione del documento.	Luca Di Chiara
16/12/2018	SDD 1.1	Revisione e modifiche minori effettuate.	Luca Di Chiara, Emilio Mainardi



07/01/2019	SDD 2.1	Aggiunta relazione "Messaggio", rimosso "Modifica orario di ricevimento", "Modifica corso" dove presente e cambiato "Aggiungi corso" in "Associa corso". Modifiche minori effettuate.	Luca Di Chiara.
------------	---------	---	-----------------



## Sommario

<b>1. Introduzione</b>	5
1.1 Obiettivi del sistema	5
1.2 Design goals	5
1.3 Definizioni, acronimi e abbreviazione	7
1.4 Riferimenti	9
<b>2. Architettura del sistema corrente</b>	9
<b>3. Architettura del sistema proposto</b>	9
3.1 Panoramica	9
3.2 Decomposizione in sottosistemi	10
3.2.1 <i>Decomposizione in Layer</i>	10
3.2.2 <i>Decomposizione in sottosistemi</i>	10
3.2.3 <i>Diagramma di Deployment</i>	11
3.3 Mapping Hardware/Software	12
3.4 Gestione dei dati persistenti	12
3.5 Controllo degli accessi e sicurezza	15
3.6 Controllo flusso globale del sistema	15
3.7 Condizione limite	15
3.7.1 <i>Start-up</i>	15
3.7.2 <i>Start-up (a seguito di un fallimento)</i>	16
3.7.3 <i>Terminazione</i>	16
3.7.4 <i>Fallimento</i>	16
<b>4. Servizi dei sottosistemi</b>	17



## 1. Introduzione

### 1.1 Obiettivi del sistema

Lo scopo di questo sistema è realizzare un sistema software che permetta la gestione dei dati e delle prenotazioni dei ricevimenti della facoltà di Informatica dell'Università degli Studi di Salerno e di consentire una facile gestione di quest'ultime, il tutto supportato dalla creazione di un portale web accessibile anche da mobile.

Lo sviluppo di una piattaforma di questo tipo permetterebbe di realizzare uno strumento in grado di ottimizzare la gestione dei ricevimenti e dell'organizzazione tra docente-studente, dando la possibilità agli studenti di prenotarsi e al docente di gestirle in maniera più semplice.

La piattaforma dovrebbe quindi consentire un'organizzazione più immediata ed efficace tra gli attori del sistema (studente, docente).

La piattaforma è progettata affinché uno studente possa prenotarsi ad un determinato orario di ricevimento se libero, e per il docente una gestione di tutte le prenotazioni effettuate visualizzandole nel dettaglio e, nel caso, annullarle per problematiche varie. Inoltre, permette allo studente di poter ricercare un determinato docente e di poterlo inserire tra i preferiti in modo tale da poter effettuare una prenotazione in maniera più rapida senza ricarlo, accedendo ad un'area impiegata ad hoc.

Infine, il docente potrà modificare la sua area personale, inserendo nuovi orari di ricevimento e eliminarne di vecchi; oppure aggiungere, modificare e eliminare corsi ad esso associati.

Dato che il sistema ha accesso a dati sensibili di studenti e docenti, deve fornire un metodo di autenticazione sicuro in modo che i dati siano protetti da accessi fraudolenti.

Inoltre per una migliore usabilità, il sistema dovrà essere facile da apprendere ed intuitivo da utilizzare, deve consentire la navigazione agevole per la fruizione delle funzionalità da lui offerte, ridurre la documentazione utente al minimo, permettere un utilizzo agevole del sistema anche senza consultare la documentazione.

### 1.2 Design goals

#### *Criteri di performance*

- Tempo di risposta  
Per la visualizzazione di una determinata prenotazione il tempo di risposta sarà 3 secondi.  
Per la visualizzazione della lista degli studenti prenotati il tempo di risposta sarà 5 secondi.  
Per la ricerca di un professore, da parte dello studente, il tempo di risposta sarà 3 secondi.  
Per la modifica e l'aggiunta di un elemento al database, il tempo di risposta sarà 5 secondi.
- Memoria  
La dimensione complessiva del sistema dipenderà dalla memoria utilizzata per il mantenimento del database

#### *Criteri di affidabilità*

- Robustezza  
Eventuali input non validi immessi dall'utente saranno opportunamente segnalati con un messaggio di errore.



- Affidabilità

I risultati visualizzati saranno attendibili. Per quanto riguarda i docenti, rispecchieranno la reale situazione all'interno della facoltà di Informatica dell'Università degli studi di Salerno. Ad esempio, saranno visibili gli orari di ricevimento nelle loro pagine personali. I ricevimenti saranno resi disponibili nell'apposita sezione, e rappresenteranno le informazioni relative al colloquio richiesto. I messaggi, disponibili per il docente ed utili per comunicare la sua eventuale assenza, saranno inviati a tutti gli studenti prenotati per il ricevimento.

- Disponibilità

Una volta online, il sistema sarà fruibile da tutti gli studenti e i docenti registrati.

- Tolleranza ai guasti

Il sistema potrà subire guasti dovuti al sovraccarico del database con successivo fallimento. Per ovviare al problema, periodicamente sarà previsto un salvataggio dei dati sotto forma di codice SQL necessario per la rigenerazione del database.

- Security

L'accesso al sistema sarà garantito mediante una e-mail e una password.

### ***Criteri di costo***

- Costi di sviluppo

È stimato un costo complessivo di 400 ore per la progettazione e lo sviluppo del sistema (50 ore per ogni membro del team)

### ***Criteri di manutenzione***

- Estendibilità

Se modificato adeguatamente, il sistema potrà essere esteso ad altre Università. Inoltre, potranno essere aggiunti nuovi parametri riguardanti gli studenti, i docenti e le prenotazioni.

- Adattabilità

Il sistema potrà essere utilizzato da altre facoltà ed Università.

- Tracciabilità dei requisiti

La tracciabilità dei requisiti sarà possibile grazie ad una matrice di tracciabilità, attraverso la quale sarà possibile retrocedere al requisito associato ad ogni parte del progetto. La tracciabilità sarà garantita dalla fase di progettazione fino al testing.

- Portabilità

Il sistema sarà portabile in quanto l'interazione avviene mediante un browser senza interazione con il sistema sottostante. Vi è quindi indipendenza dal sistema operativo.

### ***Criteri utenti finali***

- Usabilità

Il sistema sarà molto semplice da apprendere senza la consultazione di documentazione associata. L'intuitività sarà garantita in quanto il sistema avrà una buona prevedibilità, cioè la risposta del sistema ad un'azione utente sarà corrispondente alle aspettative.

- Utilità

Il lavoro dell'utente verrà supportato nel miglior modo possibile dal sistema, infatti l'utente compirà le operazioni consentite senza il carico di lavoro che deriva dal realizzare gli stessi procedimenti svolti con mezzi cartacei.

***Priorità Design Goals***

Design Goals	Priorità
Tempo di risposta	Alta
Memoria	Media
Robustezza	Alta
Affidabilità	Alta
Disponibilità	Alta
Tolleranza ai guasti	Alta
Security	Alta
Costi di sviluppo	Media
Estendibilità	Media
Adattabilità	Media
Tracciabilità dei requisiti	Alta
Portabilità	Alta
Usabilità	Alta
Utilità	Alta

### 1.3 Definizioni, acronimi e abbreviazione

**Studio:** rappresenta lo studio di un docente, nonché luogo di incontro tra docenti e studenti.

**Studente:** rappresenta uno studente universitario, ognuno avente un proprio account in cui saranno specificate le seguenti informazioni:

1. **Nome;**
2. **Cognome;**
3. **E-mail istituzionale;**
4. **Matricola;**
5. **Password.**

**Docente:** rappresenta un docente universitario, ognuno avente un proprio account in cui saranno specificate le seguenti informazioni:

1. **Nome;**
2. **Cognome;**



3. **E-mail istituzionale;**
4. **Corsi;**
5. **Calendario ricevimento;**
6. **Password;**
7. **Studio,**
8. **Immagine.**

**Corso:** rappresenta un corso universitario tenuto da uno o più docenti.

**Prenotazione:** rappresenta una prenotazione di ricevimento effettuata da uno studente.

**Ricevimento:** rappresenta l'accoglienza di uno studente da parte di un docente dopo aver effettuato la prenotazione.

### ***Acronimi***

SM = Safe Meeting

RAD = Requirement Analysis Document.

RF = Requisito Funzionale.

RNF = Requisito Non Funzionale.

SC = Scenario.

UC = Use Case.

UCD = Use Case Diagram.

CD = Class Diagram.

SD = Sequence Diagram.

SCD = Statechart Diagram.

NP = Navigation Path.

MU = Mock-up.

MVC = Model View Controller.

SQL = Structured Query Language.

FURPS+ = Rappresenta:

- Funzionalità;
- Usabilità;
- Affidabilità;
- Prestazioni;
- Supportabilità.

Il “+” sta per pseudo-requisiti o vincoli del sistema:

- Implementazione;
- Interfaccia;
- Operazioni;
- Packaging;
- Legali.

AUT = Autenticazione.

ACC = Account.

STUD = Studente.





DOC = Docente.

NA = Nessuna.

## 1.4 Riferimenti

- <http://www.safemeeting.it>
- Bernd Bruegge & Allen H. Dutoit, *Object-Oriented Software Engineering: Using UML, Patterns and Java*, (2nd edition), Prentice-Hall, 2004.
- Ian Sommerville, *Ingegneria del software*, (8a edizione), Pearson, 2007.
- RAD\_SM\_v3.0.

## 2. Architettura del sistema corrente

Attualmente la gestione dei ricevimenti universitari non è gestita tramite un sistema software. Questi, infatti, sono organizzati e gestiti tramite documenti cartacei o tramite lo scambio di mail tra i docenti e gli studenti interessati. Tale organizzazione provoca non pochi disagi agli studenti e soprattutto uno spreco di risorse che potrebbero essere utilizzate diversamente. Si tratta quindi di un sistema che rientra nel campo della Greenfield Engineering. Infatti in un Greenfield Project lo sviluppo comincia da zero, non esiste nessun sistema a priori e i requisiti sono ottenuti dall'utente finale e dal cliente. Nasce, perciò, a partire dai bisogni dell'utente.

## 3. Architettura del sistema proposto

### 3.1 Panoramica

Il sistema da noi proposto è un'applicazione web gestita in locale per l'Università degli studi di Salerno. L'obiettivo che si pone è quello di fornire uno strumento per la gestione interna dei ricevimenti.

Il sistema è rivolto ai docenti e agli studenti della facoltà di informatica. Ogni utente potrà accedervi tramite una login. In base alla tipologia di utente (docente o studente) si avrà una visione diversa del sistema.

Lo studente avrà la possibilità di ricercare i docenti, avere una lista di preferiti e fissare un ricevimento, sarà possibile visualizzare la prenotazione e, se necessario, eliminarla.

Il docente, a sua volta, potrà visualizzare la lista delle prenotazioni, potrà indicare un'eventuale assenza in studio, gestire i corsi e gli orari.

Il sistema verrà suddiviso in client e server, il client gestirà la parte di presentazione e la parte di logica direttamente connessa all'interfaccia grafica, il server invece gestirà la parte di logica relativa ai dati, che saranno memorizzati in un database salvato sul server. Per motivi di sicurezza, in quanto verranno gestiti dati sensibili, tutto il sistema sarà su una sola macchina.

I gestori saranno individuati in base alle funzionalità per poter rendere massima la coesione e minimo l'accoppiamento tra i sottosistemi in modo che i cambiamenti in un sottosistema non influiscano sugli altri.

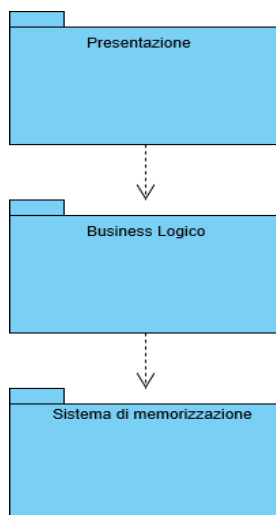
Le funzionalità saranno divise in layer logici in base alle differenti funzionalità utilizzando il modello MVC: sistema di memorizzazione (Model), presentazione (View) e business logic (Controller).

## 3.2 Decomposizione in sottosistemi

### 3.2.1 Decomposizione in Layer

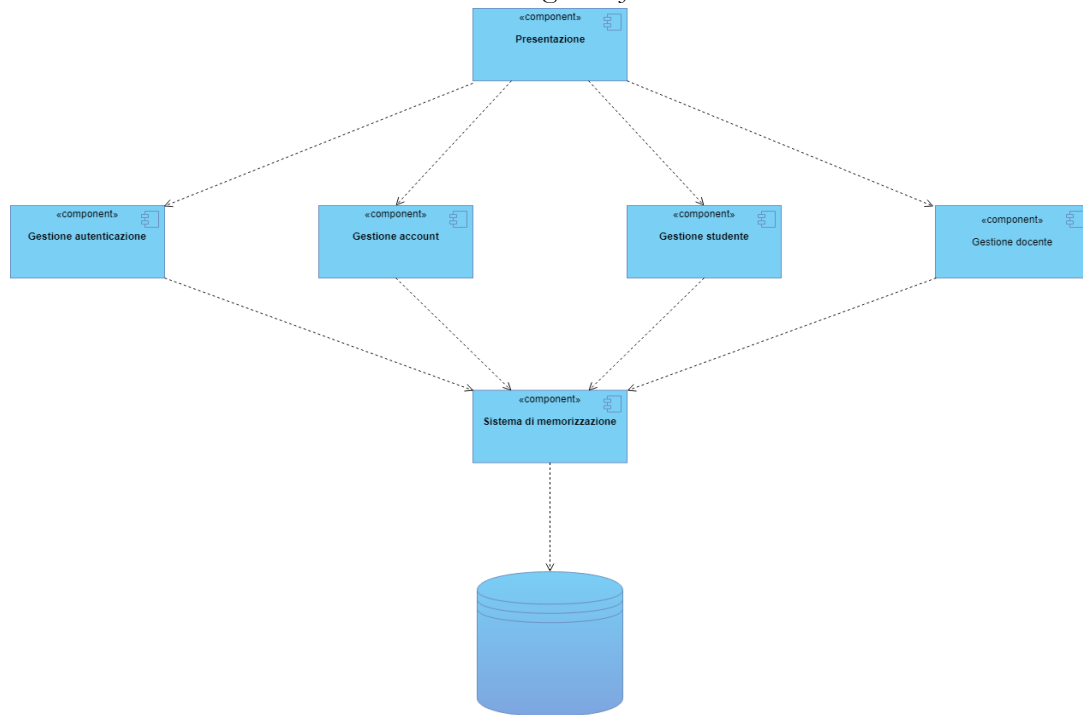
La decomposizione prevista per il sistema è composta da tre layer che si occupano di gestire aspetti e funzionalità differenti:

- sistemi di memorizzazione (Model): si occupa della gestione e dello scambio dei dati tra i sottosistemi;
- presentazione(View): raccoglie e gestisce l'interfaccia grafica e gli eventi generati dall'utente;
- business logico (Controller): si occupa della gestione della logica del sistema;



### 3.2.2 Decomposizione in sottosistemi

Dopo un'attenta e dettagliata analisi funzionale, abbiamo scelto di dividere il nostro sistema nel seguente modo in quanto, per la divisione in componenti, avevamo bisogno di un basso accoppiamento e un'elevata coesione tra i servizi offerti tra i componenti interni. Abbiamo deciso di suddividere le funzionalità per area di gestione e creato un'interfaccia intermedia tra i sistemi della logica di business e il database, in modo da non dover fare grossi cambiamenti nel caso in cui sia necessario cambiare database. Il sistema si compone di sei componenti che si occupano di gestire aspetti e funzionalità differenti:



Gestione account e Gestione autenticazione sono riferite alle funzionalità dell'utente generico;  
Gestione studente è riferita alle funzionalità dello studente;  
Gestione docente è riferita alle funzionalità del docente.

Il livello Presentazione prevede un unico sottosistema:

- Presentazione: sistema che gestisce l'interfaccia grafica e gli eventi generati dall'interazione con il sistema.

Il livello Business Logico prevede a sua volta una suddivisione in 4 sottosistemi:

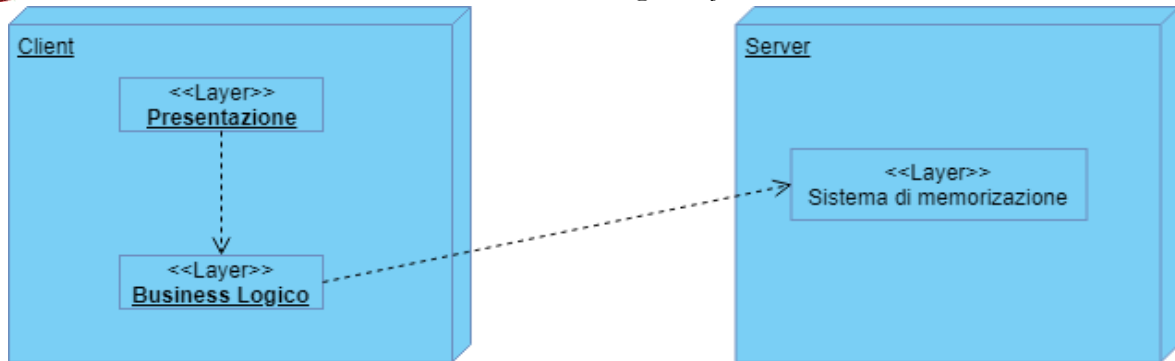
- Gestione autenticazione;
- Gestione account;
- Gestione studente;
- Gestione docente.

Il livello Sistema di memorizzazione prevede un unico sottosistema:

- Sistema di memorizzazione: sistema che gestisce ed immagazzina i dati persistenti.

### ***3.2.3 Diagramma di Deployment***

L'utente richiede le funzionalità tramite l'interfaccia che il sistema mette a disposizione. Il client ospita lo strato di presentazione e anche la parte server che ingloba logica del sistema e gestisce la persistenza dei dati.



### 3.3 Mapping Hardware/Software

Il sistema che si vuole sviluppare sarà diviso in client e server che saranno installati su Tomcat ed utilizzerà un DBMS MySQL.

### 3.4 Gestione dei dati persistenti

*Legenda priorità:*

Obbligatorio	
Trascurabile	

*Docente*

PARAMETRO	PRIORITA'
Matricola	
Nome	
Cognome	
Email	
Password	
Studio	
Immagine	

*Corso*

PARAMETRO	PRIORITA'
Nome	
Numero cfu	
Ore totali	

*Studente*

PARAMETRO	PRIORITA'
Matricola	
Nome	
Cognome	
Email	
Password	

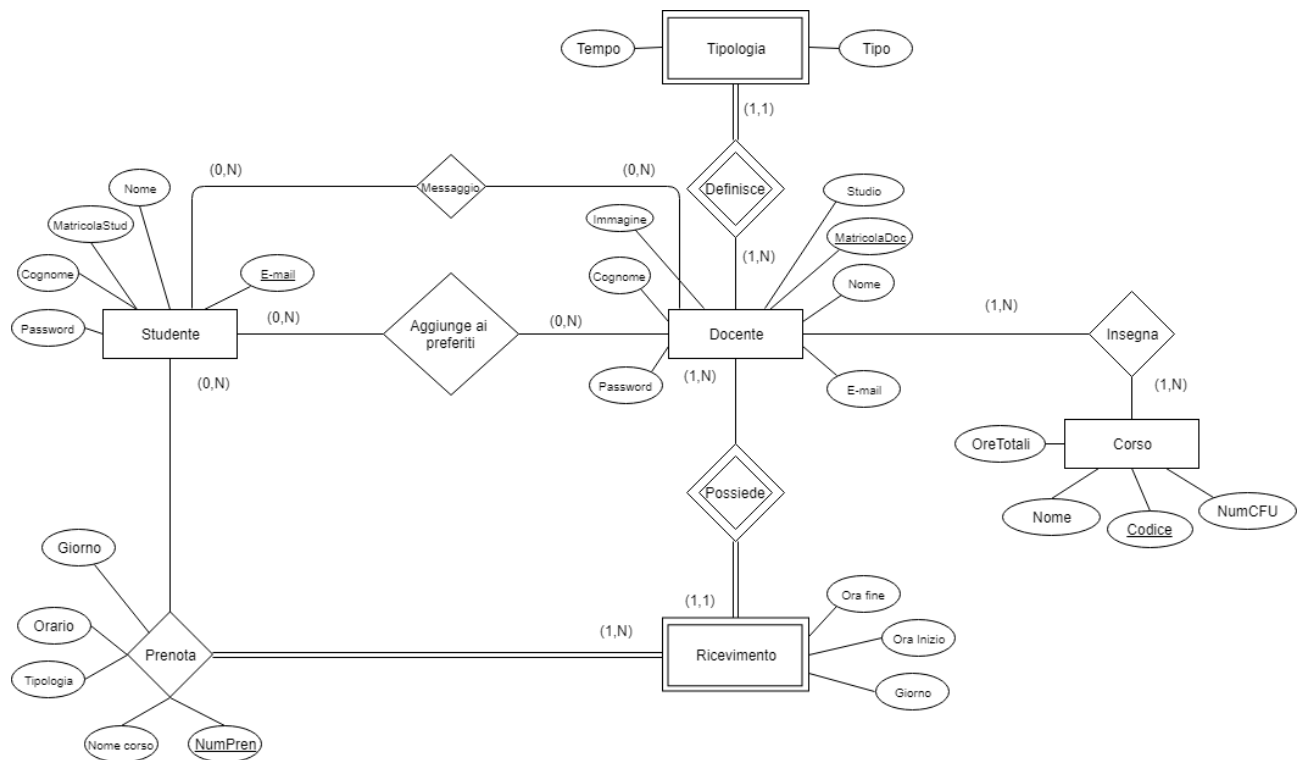
*Ricevimento*

PARAMETRO	PRIORITA'
Giorno	
Ora inizio	
Ora fine	
Tempo	

*Tipologia*

PARAMETRO	PRIORITA'
Tempo	
Tipo	

## Schema E-R



## Specifica delle caratteristiche degli attributi

### Docente

Nome	Tipo	Null	Key
MatricolaDoc	Char(10)	Not null	Primary
Nome	Varchar(50)	Not null	
Cognome	Varchar(50)	Not null	
Password	Varchar(30)	Not null	
E-mail	Varchar(50)	Not null	
Studio	Varchar(5)	Not null	
Immagine	Varchar(100)	Null	

### Studente

Nome	Tipo	Null	Key
E-mail	Varchar(50)	Not null	Primary
Nome	Varchar(50)	Not null	
Cognome	Varchar(50)	Not null	
Password	Varchar(30)	Not null	
MatricolaStud	Char(10)	Not null	



***Ricevimento***

Nome	Tipo	Null	Key
ricevMatricolaDoc	Char(10)	Not null	Foreign
Ora_inizio	Time	Not null	
Ora_fine	Time	Not null	
Giorno	Varchar(15)	Not null	

***Corso***

Nome	Tipo	Null	Key
Codice	Char(10)	Not null	Primary
Num_CFU	Int(5)	Not null	
Nome	Varchar(50)	Not null	
Ore_totali	Int(5)	Not null	

***Prenota***

Nome	Tipo	Null	Key
Numero_Prenotazione	Int	Not null	Primary
Nome_Corso	Varchar(50)	Not null	
Tipologia	Varchar(30)	Not null	
Orario	Time	Not null	
Giorno	Date	Not null	
prenotaEmailStud	Varchar(50)	Not null	Foreign
prenotaMatricolaDoc	Char(10)	Not null	Foreign

***Insegna***

Nome	Tipo	Null	Key
insegnaMatricolaDoc	Char(10)	Not null	Foreign
insegnaCodiceCorso	Char(10)	Not null	Foreign

***Aggiungi\_ai\_preferiti***

Nome	Tipo	Null	Key
preferitiMatricolaDoc	Char(10)	Not null	Foreign
preferitiEmailStud	Varchar(50)	Not null	Foreign

***Tipologia***

Nome	Tipo	Null	Key
tipoMatricolaDoc	Char(10)	Not null	Foreign
Tempo	Int(5)	Not null	
Tipo	Varchar(50)	Not null	



### Messaggio

Nome	Tipo	Null	Key
messaggioEmailStud	Char(10)	Not null	Foreign
messaggioMatricolaDoc	Char(10)	Not null	Foreign
Messaggio	Varchar(500)	Not null	

### 3.5 Controllo degli accessi e sicurezza

Il controllo degli accessi è garantito tramite l'utilizzo di e-mail e password per il docente, o lo studente, che verranno richieste per ogni singolo accesso alla piattaforma.

Le operazioni che l'utente registrato può fare sono:

SOTTOSISTEMA	FUNZIONI	ATTORE	
		Docente	Studente
Gestione Autenticazione	Login	✓	✓
	Logout	✓	✓
Gestione Account	Registra account	✓	✓
	Modifica account	✓	✓
	Elimina account	✓	✓
Gestione Ricevimenti	Ricerca docente	✗	✓
	Prenotazione	✗	✓
	Visualizza prenotazione	✗	✓
	Elimina prenotazione	✗	✓
	Aggiungi docente ai preferiti	✗	✓
	Rimuovi docente dai preferiti	✗	✓
	Associa corso	✓	✗
	Elimina corso	✓	✗
	Aggiunta orario di ricevimento	✓	✗
	Elimina orario di ricevimento	✓	✗

### 3.6 Controllo flusso globale del sistema

Il sistema SM fornisce funzionalità che richiedono una continua interazione da parte dell'utente, per questo motivo abbiamo scelto come meccanismo di controllo del flusso globale l'event-driven control.

### 3.7 Condizione limite

#### 3.7.1 Start-up

Per il primo start-up del sistema "Safe Meeting" è necessario l'avvio di un web server che fornisca il servizio di un Database MySQL per la gestione dei dati persistenti e l'interpretazione ed esecuzione del codice lato server. In seguito, tramite l'interfaccia di Login, sarà possibile autenticarsi tramite opportune credenziali (e-mail e password) come utente con accesso alle



funzionalità del sistema. Una volta effettuato l'accesso, “Safe Meeting” presenterà all'utente la home, dalla quale si possono effettuare tutte le operazioni che il sistema fornisce.

### ***3.7.2 Start-up (a seguito di un fallimento)***

Il sistema può subire guasti dovuti al sovraccarico del database con successivo fallimento. Per ovviare al problema, periodicamente è previsto un salvataggio dei dati sotto forma di codice SQL. All'avvio a seguito di tale fallimento, oltre alle normali procedure previste per lo start-up, l'ultimo codice SQL memorizzato sarà eseguito per la rigenerazione del database.

### ***3.7.3 Terminazione***

Al momento della chiusura dell'applicativo si ha la terminazione del sistema con un regolare Logout dal sistema. Viene assicurata la consistenza dei dati, annullando eventuali operazioni che erano in esecuzione.

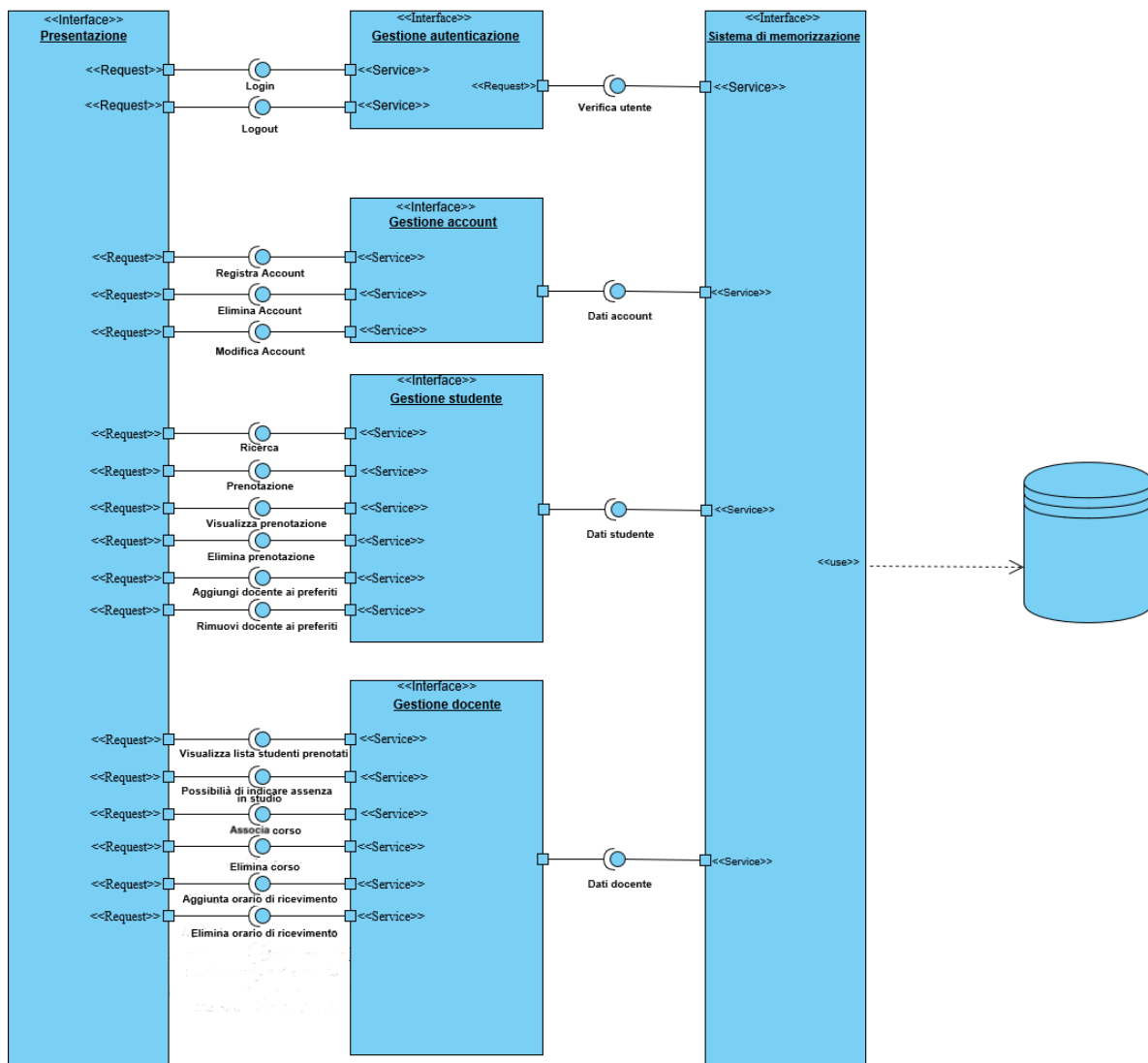
### ***3.7.4 Fallimento***

Possono verificarsi diversi casi di fallimento del sistema:

- Nel caso di malfunzionamento del database, sono previsti dei periodici backup per ripristinarlo.
- Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione, non sono previsti metodi che ripristinino lo stato del sistema a prima dello spegnimento inaspettato.
- Un altro caso di fallimento potrebbe essere dovuto ad un errore critico nell'hardware, non è prevista alcuna misura correttiva.
- Se si verifica un malfunzionamento del software, i dati modificati ma non ancora salvati nel database, andranno persi.



## 4. Servizi dei sottosistemi



**Presentazione:** sistema che gestisce l'interfaccia grafica e gli eventi generati dall'interazione con il sistema.

**Gestione autenticazione** offre 2 servizi all'interfaccia di Presentazione:

- Login
- Logout

**Gestione account** offre 3 servizi all'interfaccia di Presentazione:

- Registra account
- Elimina account
- Modifica account

**Gestione studente** offre 6 servizi all'interfaccia di Presentazione:

- Ricerca



- Prenotazione
- Visualizza prenotazione
- Elimina prenotazione
- Aggiungi docente ai preferiti
- Rimuovi docente ai preferiti

**Gestione docente** offre 6 servizi all'interfaccia di Presentazione:

- Visualizza lista studenti prenotati
- Possibilità di indicare assenza in studio
- Associa corso
- Elimina corso
- Aggiunta orario di ricevimento
- Elimina orario di ricevimento

**Sistema di memorizzazione** offre un unico servizio a Gestione autenticazione:

- Verifica utente

**Sistema di memorizzazione** offre un unico servizio a Gestione account:

- Dati account

**Sistema di memorizzazione** offre un unico servizio a Gestione studente:

- Dati studente

**Sistema di memorizzazione** offre un unico servizio di Gestione docente:

- Dati docente