



TutoratoSmart

SDD System Design Document

TutoratoSmart

Riferimento	
Versione	0.4
Data	06/12/2019
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Marco Delle Cave, Francesco Pagano, Manuel Pisciotta, Alessia Olivieri
Approvato da	



Revision History

Data	Versione	Cambiamenti	Autori
27/11/2019	0.1	Aggiunta introduzione (Obiettivi del sistema, design goals, definizioni, acronimi, abbreviazioni, riferimenti)	Delle Cave Marco, Pagano Francesco
28/11/2019	0.2	Aggiunta architettura del sistema corrente, sistema proposto, decomposizione in sottosistemi, mapping hardware-software	Delle Cave Marco, Pagano Francesco
3/12/2019	0.3	Aggiunta gestione dati persistenti, controllo degli accessi e sicurezza, controllo globale del software, condizioni limiti, servizi dei sottosistemi	Delle Cave Marco, Pagano Francesco
4/12/2019	0.4	Revisione schema ER, architettura sistema proposto	Delle Cave Marco, Pagano Francesco
6/12/2019	0.4	Revisione SDD	Pisciotta Manuel



Sommario

I.	1. Introduzione	3
	1.1 Obiettivi del sistema	3
	1.2 Design Goals	3
	1.3 Definizioni, acronimi e abbreviazioni.....	5
	1.4 Riferimenti	5
	1.5 Panoramica.....	5
II.	2. Architettura del Sistema corrente	5
III.	3. Architettura del Sistema proposto	5
	3.1 Panoramica.....	5
	3.2 Decomposizione in sottosistemi	6
	3.2.1 Decomposizione in Layer	6
	3.2.2 Decomposizione in Sottosistemi	6
	3.2.3 Diagramma di Deployment	7
	3.3 Mapping hardware/software.....	8
	3.4 Gestione dati persistenti	9
	3.5 Controllo degli accessi e sicurezza	12
	3.6 Controllo flusso globale del sistema	13
	3.7 Condizione limite.....	14
	3.7.1 Start-up	14
	3.7.2 Start-up (a seguito di un fallimento)	14
	3.7.3 Terminazione	14
	3.7.4 Fallimento.....	14
IV.	4. Servizi dei Sottosistemi	15

1. Introduzione

1.1 Obiettivi del sistema

Il progetto nasce per fornire uno strumento di supporto agli studenti e ai tutor di Psicologia dell'Università degli studi della Campania "Luigi Vanvitelli", e alla Commissione Tutorato per consentire un processo rapido, senza perdita di informazioni ed efficiente. Attualmente la prenotazione di un appuntamento allo sportello di tutorato risulta molto lenta e poco pratica in quanto avviene tramite scambio di email tra studente e tutor. Invece, per quanto riguarda l'acquisizione e la consegna finale dei registri dell'attività di tutorato, è necessario rivolgersi fisicamente alla Segreteria di Dipartimento e, per tutta la durata del contratto, ogni tutor deve annotare sul proprio registro cartaceo le attività svolte, attività che successivamente verranno valutate e convalidate dalla Commissione di Tutorato. Questo crea molti disagi e rallentamenti nelle pratiche.

Al fine di facilitare l'iter burocratico sopra descritto, gli obiettivi primari del sistema sono quindi quelli di:

1. Fornire uno strumento per supportare la prenotazione allo sportello di tutorato e la gestione delle attività dei tutor stessi;
2. Migliorare lo scambio di informazioni tra i tutor e gli altri stakeholder coinvolti;
3. Ottimizzare i tempi relativi alla prenotazione allo sportello di tutorato e al riconoscimento delle ore lavorative svolte;
4. Eliminare gli attuali disagi agli stakeholder.

1.2 Design Goals

I design goal identificati per il sistema TutoratoSmart sono i seguenti:

- **Criteri di performance**
 - o Tempo di risposta
 - Per l'accesso all'area utente il tempo di risposta è di 2 secondi.
 - Per la visualizzazione dei form di richiesta il tempo di risposta è di 1 - secondo.
 - Per la generazione del file Excel il tempo di risposta è di 3 secondi.
 - Per la visualizzazione delle liste il tempo di risposta è di 3 secondi.
 - Per la modifica di appuntamento, il tempo di risposta è di 1 secondo.
 - o Memoria
 - La dimensione complessiva del sistema dipende dalla memoria utilizzata per il mantenimento del database.
- **Criteri di affidabilità**
 - o Robustezza
 - Eventuali input non validi immessi dall'utente saranno opportunamente segnalati attraverso messaggi di errore.
 - o Affidabilità
 - I risultati visualizzati sono attendibili e rispecchiano la reale situazione all'interno del Dipartimento di Psicologia in funzione dell'aggiornamento dei dati. Qualora l'utente richieda la visualizzazione di dati in un particolare periodo di tempo, i valori saranno calcolati in funzione dei dati relativi a tale intervallo di tempo.



- Disponibilità
 - Una volta realizzato il sistema, sarà disponibile ogni qualvolta gli studenti, i tutor e la commissione tutorato ne richiederanno l'utilizzo dato che il sistema è in locale.
- Tolleranza ai guasti
 - 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 necessario per la rigenerazione del database.
- Security
 - L'accesso al sistema è garantito mediante una email e una password. La sicurezza del database è garantita dal fatto che lo stesso è in locale, quindi accessibile solo all'addetto ai lavori.
- Criteri di costo
 - Costi di sviluppo
 - È stimato un costo complessivo di 200 ore per la progettazione e lo sviluppo del sistema (50 ore per ogni team member).
- Criteri di manutenzione
 - Estendibilità
 - Il sistema è estendibile in quanto esso stesso può essere esteso ad altri dipartimenti o ad altre università cambiando i dati su cui lavora. Inoltre è possibile aggiungere altri indicatori o altre modalità di visualizzazione mediante limitata quantità di codice da integrare.
 - Adattabilità
 - Il sistema può funzionare solo in ambito universitario, ma è adattabile a più dipartimenti modificando i dati su cui lavora.
 - Tracciabilità dei requisiti
 - La tracciabilità dei requisiti è possibile grazie ad una matrice di tracciabilità, attraverso la quale è possibile retrocedere al requisito associato ad ogni parte del progetto. La tracciabilità è 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, c'è quindi indipendenza dal sistema operativo.
- Criteri utenti finali
 - Usabilità
 - L'usabilità di un sistema può essere analizzata considerando diverse caratteristiche. Questo sistema sarà molto semplice da apprendere senza la consultazione di documentazione associata. L'intuitività è 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 calcoli manualmente.

1.3 Definizioni, acronimi e abbreviazioni

RAD: Requirements Analysis Document.

SDD: System Design Document.

ODD: Object Design Document.

DB: Database.

GREENFIELD ENGINEERING: Tipologia di sviluppo che 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.

1.4 Riferimenti

- Ian Sommerville, *Software Engineering*, Addison Wesley.
- TS_RAD_1.0

1.5 Panoramica

Al secondo punto del documento verrà presentato il sistema corrente.

Al terzo punto verrà presentata l'architettura del sistema proposto in cui gestiremo la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema, le condizioni limite.

Al quarto punto verranno presentati i servizi dei sottosistemi.

2. Architettura del Sistema corrente

Attualmente non esiste un sistema software che si occupa di gestire questa problematica, ossia la gestione degli appuntamenti di tutorato. Gli appuntamenti attualmente vengono gestiti tramite scambi di email tra studenti e tutor, e ciò risulta essere molto macchinoso e alquanto lento poiché i tempi necessari allo scambio di email e il recupero dei dati da parte dello studente non sono agevoli. Quindi si tratta 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 con lo scopo di offrire un supporto alle attività di tutorato. L'obiettivo che si pone a fornire uno strumento di gestione di richieste di tutorato sia dal lato studente che dal lato tutor. Metterà a disposizione dello studente le interfacce per registrarsi alla piattaforma, per prenotare un nuovo appuntamento e per modificarlo; mentre invece il tutor potrà accettare o spostare gli appuntamenti, confermare questi ultimi inserendo ulteriori dettagli e compilare il registro del tutorato inserendo le attività lavorative svolte, sempre attraverso delle interfacce grafiche. Verrà utilizzata un'architettura di tipo MVC, dove il model fornirà le operazioni

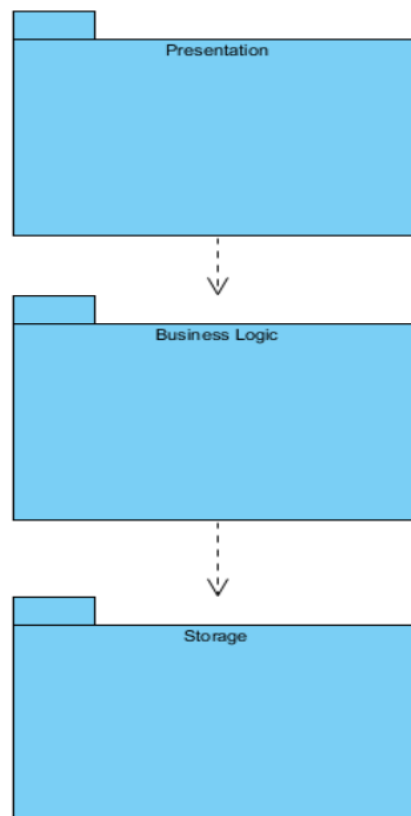
per accedere ai dati utili all'applicazione, ed implementerà quindi la struttura dati centrale; il controller gestisce il control flow, ovvero ottiene gli input dall'utente e manda messaggi al Model; i View visualizzano il Model e sono notificati ogni volta che il Model è modificato. I gestori saranno 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.

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 gestirne aspetti e funzionalità differenti:

- **Presentation:** raccoglie e gestisce l'interfaccia grafica e gli eventi generati dall'utente, contiene boundary/interface;
- **Business Logic:** si occupa della gestione della logica del sistema, contiene gli oggetti control e la logica dell'applicazione;
- **Storage:** si occupa della gestione e dello scambio dei dati tra i sottosistemi, contiene gli oggetti entity del dominio di applicazione;

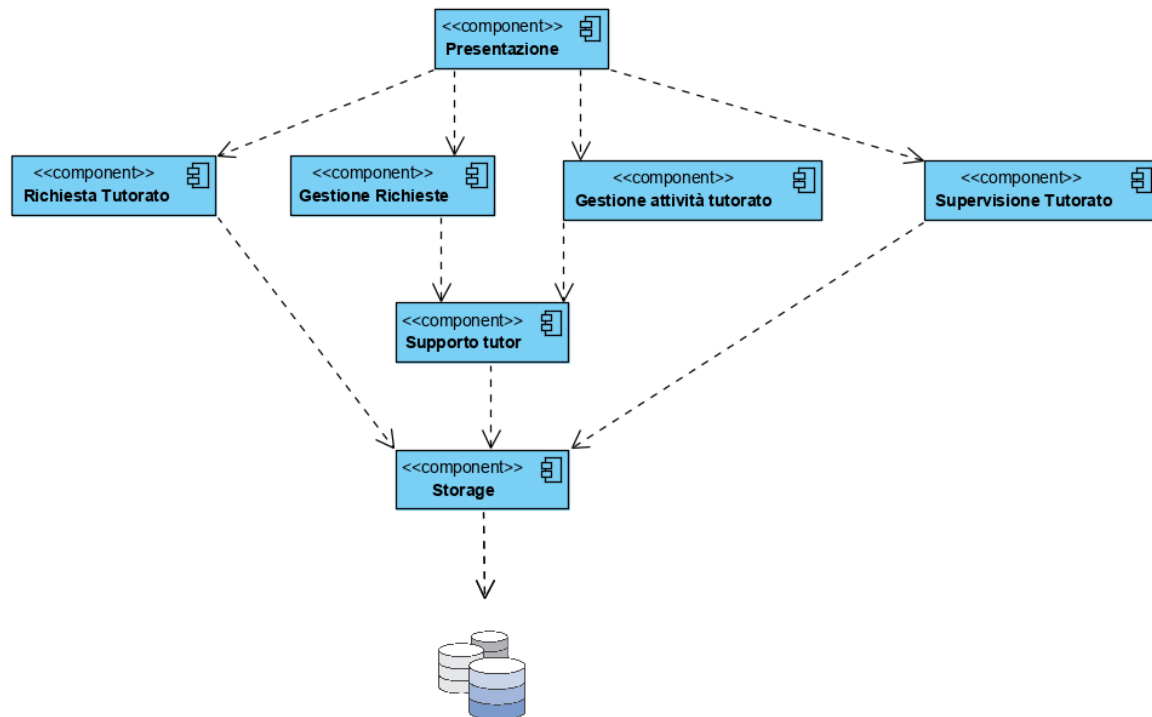


3.2.2 *Decomposizione in Sottosistemi*

Dopo un'analisi funzionale dettagliata, 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 (studente, tutor e commissione tutorato) e creato un'interfaccia intermedia storage tra i sistemi della logica di business e il database, l'assunzione dietro questa scelta

di design è che lo storage ha una interfaccia più stabile rispetto al database e quindi nel caso in cui cambi l'interfaccia del sottosistema database, solo il sottosistema storage deve cambiare

Il sistema si compone di sette componenti che si occupano di gestirne aspetti e funzionalità differenti:



Il livello Presentation prevede un unico sottosistema:

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

Il livello Business Logic prevede a sua volta una suddivisione in 5 sottosistemi e che si occupa di gestire la logica di business del sistema:

- Richiesta Tutorato;
- Gestione Richieste;
- Gestione attività tutorato;
- Supporto tutor;
- Supervisione Tutorato;

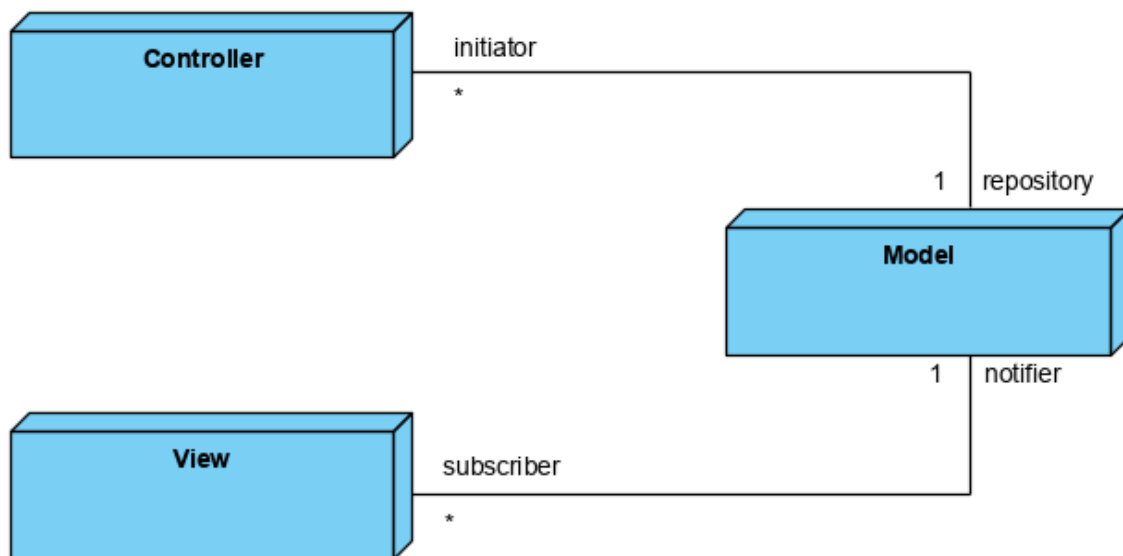
Il livello Storage prevede un unico sottosistema:

- Storage: 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 sistema sarà diviso in layer logici in base alle differenti funzionalità: presentazione, business logic e sistema di memorizzazione (dati persistenti). Utilizzeremo il pattern architetturale MVC (Model-View-Controller) basato sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

- il model mantiene la conoscenza del dominio di applicazione e fornisce le operazioni per accedere ai dati utili all'applicazione;
- il view visualizza all'utente gli oggetti del dominio dell'applicazione, visualizza i dati contenuti nel model e si occupa dell'interazione con utenti;
- il controller è responsabile della sequenza di interazioni con l'utente, riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti;

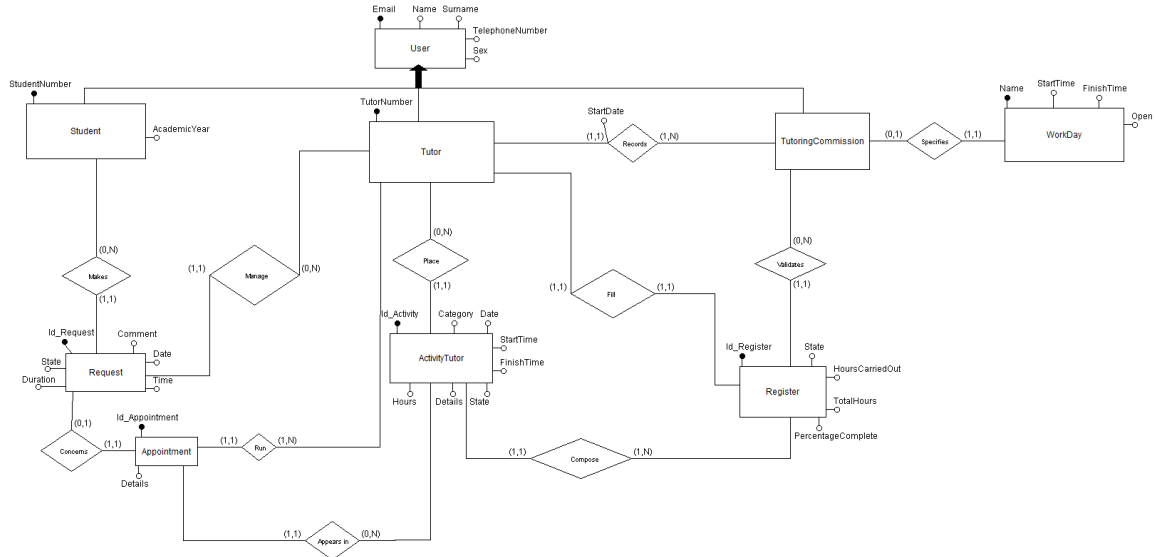


3.3 Mapping hardware/software

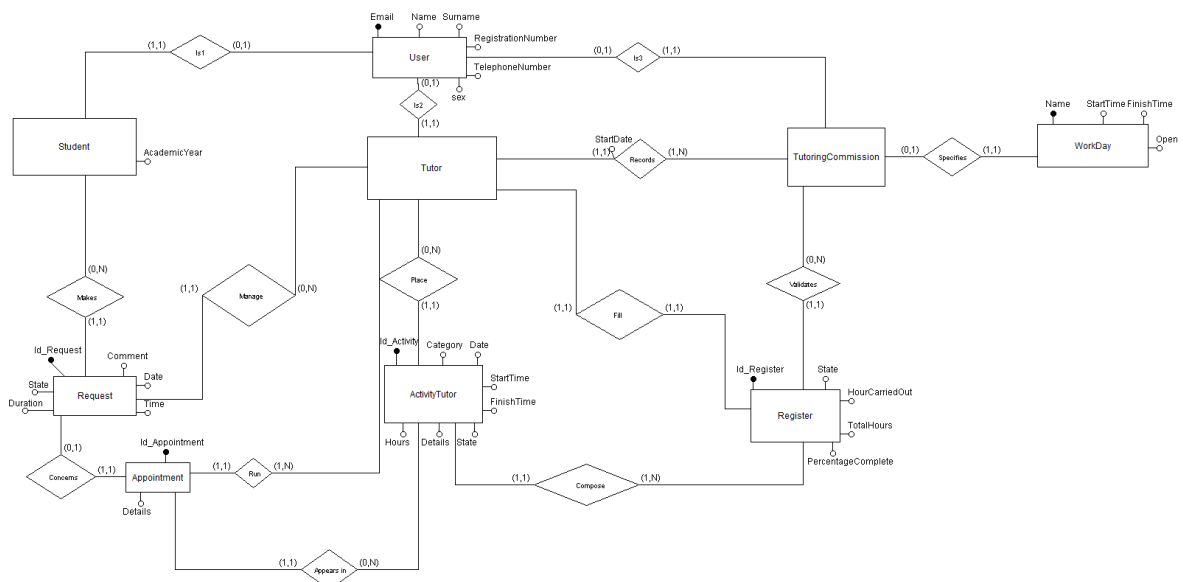
Il sistema che si vuole sviluppare sarà installato su un solo computer e utilizzerà un DBMS MySQL installato sullo stesso. Il sistema sarà diviso in client e server che inizialmente saranno lo stesso pc, ma in caso di futura necessità potranno essere facilmente divisi in quanto i servizi saranno progettati separatamente. Sul server ci sarà un DBMS per la gestione dei dati persistenti di nostro interesse.

3.4 Gestione dati persistenti

Schema ER



Schema ER ristrutturato





Mapping ER-relazionale

User(**email**, name, surname, telephoneNumber, sex, registrationNumber)

Student(**student.email**, academicYear)

Tutor(**tutor.Email**, **TutoringCommission.Email**↑, startTime)

TutoringCommission(**TutoringCommission.email**, **WorkDay** ↑,)

Request(**Id Request**, **student.email** ↑, **tutor.email** ↑, comment, date, time, state, duration)

Appointment(**Id appointment**, **tutor.email** ↑, **Activity** ↑, **Request**↑, details)

ActivityTutor(**Id activity**, **tutor.Email** ↑, **Register** ↑, category, date, startTime, finishTime, state, hours, details)

Register (**Id register**, **Tutor** ↑, **TutoringCommission** ↑, state, hoursCarriedOut, totalHours, percentageComplete)

WorkDay(**Name**, startTime, finishTime, open)

User

Nome	Tipo	Null	Key
Email	Varchar(30)	Not null	Primary key
Name	Varchar(20)	Not null	
Surname	Varchar(20)	Not null	
TelephoneNumber	Int(10)	Not null	
Sex	Char(1)	Nullable	
RegistrationNumber	Int(10)	Not null	

Student

Nome	Tipo	Null	Key
Email	Varchar(30)	Not null	Primary key/Foreign key
AcademicYear	Date	Not null	

Tutor

Nome	Tipo	Null	Key
Email	Varchar(30)	Not null	Primary key/Foreign key
StartTime	Date	Not null	



TutoringCommission

Nome	Tipo	Null	Key
Email	Varchar(30)	Not null	Primary key/Foreign key

ActivityTutor

Nome	Tipo	Null	Key
Id_Activity	Int(10)	Not null	Primary key/Foreign key
Category	Varchar(30)	Not null	
Date	Date	Not null	
StartTime	Int	Not null	
FinishTime	Int	Not null	
State	Varchar(20)	Not null	
Details	Varchar(320)	Not null	
Hours	Int	Not null	

Appointment

Nome	Tipo	Null	Key
Id_Appointment	Int(10)	Not null	Primary key
Details	Varchar(240)	Not null	

Request

Nome	Tipo	Null	Key
Id_Request	Char(10)	Not null	Primary key
Comment	Varchar(240)	Not null	
State	Varchar(10)	Not null	
Date	Date	Not null	
Time	Int	Not null	
Duration	Int	Not null	

Register

Nome	Tipo	Null	Key
Id_Register	Int(10)	Not null	Primary key/Foreign key
hoursCarriedOut	Int	Not null	



totalHours	Int	Not null	
percentageComplete	Int	Not null	
State	Varchar(10)	Not null	

WorkDay

Nome	Tipo	Null	Key
Name	varchar(10)	Not null	Primary key
StartTime	Int	Not null	
FinishTime	Int	Not null	
Open	boolean	Not null	

3.5 Controllo degli accessi e sicurezza

Il controllo degli accessi è garantito tramite l'utilizzo di email e password per lo studente, il tutor e la commissione tutorato, che verranno richieste per ogni singolo accesso.

La sicurezza sui dati sensibili degli studenti è garantita dall'accesso controllato in quanto solo la commissione tutorato può avere accesso ai dati relativi agli studenti.

Le operazioni che gli utenti possono fare sono:

Sottosistema Attori	Gestione			
	Richieste tutorato	Gestione richieste	Gestione attività tutorato	Supervisione tutorato
Studente	<ul style="list-style-type: none"> • Compilazione richiesta • Visualizzazione stato richiesta • Modifica prenotazione 	/	/	/
Tutor	/	<ul style="list-style-type: none"> • Visualizzazione dettagli richiesta • Gestione richiesta • Conferma appuntamento 	<ul style="list-style-type: none"> • Visualizzazione calendario appuntamenti • Generazione registro • Visualizzazione registro 	/

			<ul style="list-style-type: none"> • Visualizzazione attività lavorativa • Aggiunta attività lavorativa • Modifica attività lavorativa • Visualizzazione dettagli appuntamento • Modifica appuntamento 	
Commissione tutorato	/	/	/	<ul style="list-style-type: none"> • Visualizzazione student • Convalida attività tutor • Registrazione tutor • Visualizzazione tutor • Specifica giorni/orari sportello • Visualizzazione dettagli attività • Visualizzazione dettagli registro

3.6 Controllo flusso globale del sistema

Il sistema Tutorato Smart fornisce funzionalità che richiedono una continua interazione da parte dell'utente, per tal ragione abbiamo adottato un controllo del flusso globale del sistema di tipo event-driven, che è un tipo di controllo flessibile e buono per le interfacce utenti.



3.7 Condizione limite

3.7.1 Start-up

Per il primo start-up del sistema “Tutorato Smart” 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 (username e password) come utente con pieno accesso a tutte le funzionalità del sistema. Una volta effettuato l'accesso, “Tutorato Smart” presenterà all'utente la home, dal 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

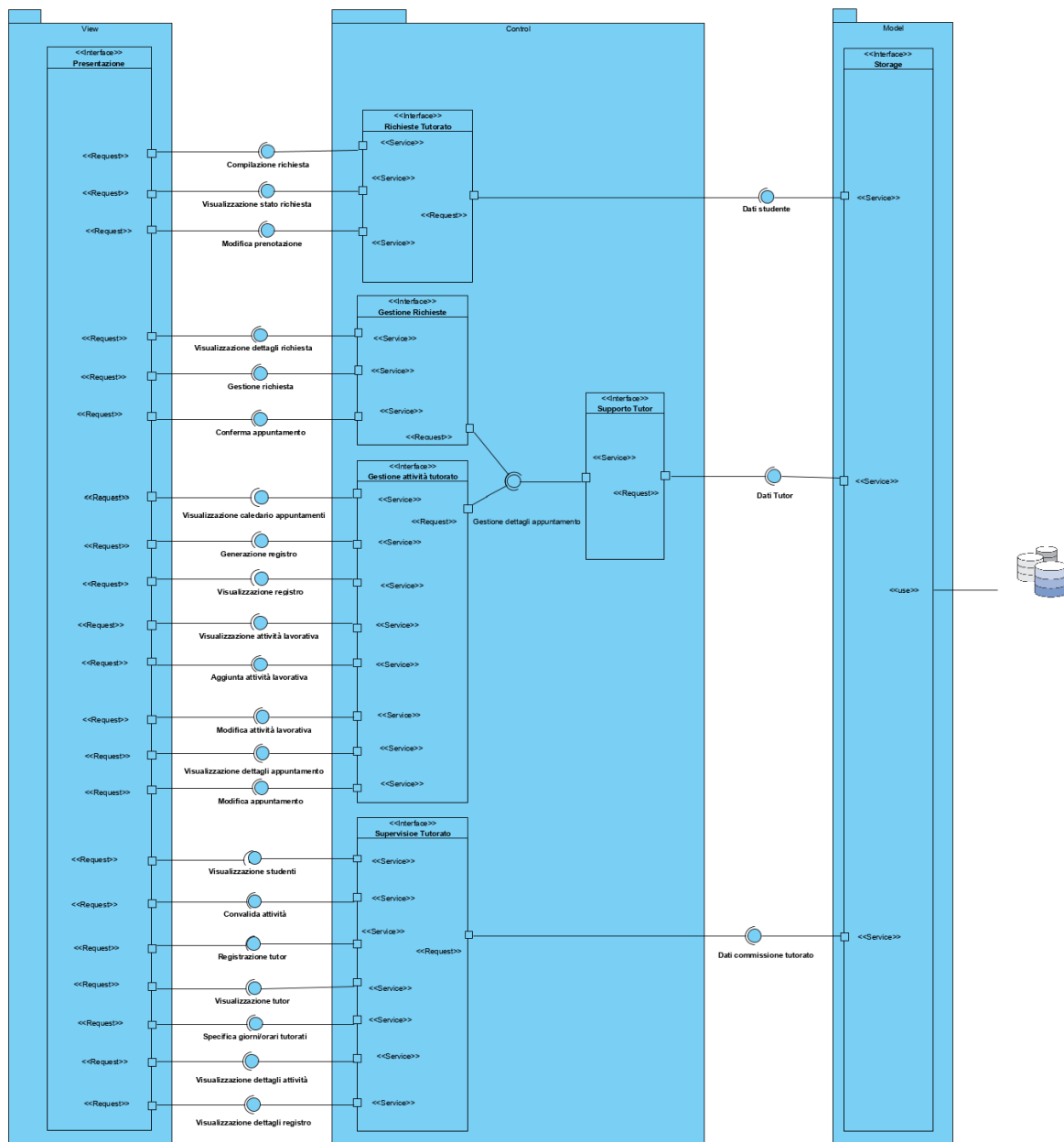
Al momento della chiusura dell'applicativo si ha la terminazione del sistema con un regolare Log-out dal sistema. Viene assicurata la consistenza dei dati, annullando eventuali operazioni che erano in esecuzione.

3.7.4 Fallimento

Possono verificarsi diversi casi di fallimento del sistema:

1. Nel caso di guasti dovuti al sovraccarico del database con successivo fallimento dello stesso, è prevista come procedura preventiva il salvataggio periodico dei dati sotto forma di codice SQL per la successiva rigenerazione del DB.
2. 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.
3. Un altro caso di fallimento potrebbe derivare dal software stesso che causa una chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione, non sono previste politiche correttive, l'unico processo che potrà essere eseguito è la chiusura del sistema e il suo successivo riavvio.
4. Un altro caso di fallimento potrebbe essere dovuto ad un errore critico nell'hardware, non è prevista alcuna misura correttiva.
5. Un altro caso di fallimento potrebbe essere dovuto ad un mancato salvataggio di dati causato da un malfunzionamento del sistema.

4. Servizi dei Sottosistemi



Presentation: Interfacce che gestiscono l'interfaccia grafica e gli eventi generati dall'interazione dell'utente con il sistema.

Richiesta Tutorato: Offre 3 servizi all'interfaccia di Presentazione:

- Compilazione richiesta
- Visualizzazione stato richiesta
- Modifica prenotazione

Gestione richieste: Offre 3 servizi all'interfaccia di Presentazione:

- Visualizzazione dettagli richiesta
- Gestione richiesta
- Conferma appuntamento



Gestione attività tutorato: Offre 8 servizi all'interfaccia di Presentazione:

- Visualizzazione calendario appuntamenti
- Generazione registro
- Visualizzazione registro
- Aggiunta attività lavorativa
- Visualizzazione attività lavorativa
- Modifica attività lavorativa
- Visualizzazione dettagli appuntamento
- Modifica appuntamento

Supporto tutor: Offre 1 servizio all'interfaccia di Presentazione:

- Gestione dettagli appuntamento

Supervisione tutorato: Offre 7 servizi all'interfaccia di Presentazione:

- Visualizzazione studenti
- Convalida attività tutor
- Registrazione tutor
- Visualizzazione tutor
- Specifica giorni/orari sportello
- Visualizzazione dettagli attività
- Visualizzazione dettagli registro

Storage offre un unico servizio a Richiesta Tutorato:

- Dati Studente

Storage offre un unico servizio a Supporto Tutor:

- Dati Tutor

Storage offre un unico servizio a Supervisione Tutorato:

- Dati Commissione Tutorato