

04/12/2018

# SDD

# SharErasmus

Share your Erasmus experience



Riferimento	
Versione	1.0
Data	04/12/2018
Destinatario	Prof.ssa F. Ferrucci
Proposto da	Federico Vitale, Francesco Vicidomini
Approvato da	



## Revision History

Data	Versione	Descrizione	Autori
25/11/2018	1.0	Stesura Introduzione e architettura del sistema corrente	Davide Bottiglieri, Paolo Cantarella, Rosaria Iorio
29/11/2018	1.0	Architettura del sistema proposto	Davide Bottiglieri, Francesco Breve, Silvio Corso, Vincenzo Sabato
02/12/2018	1.0	Servizi dei sottosistemi	Francesco Breve, Silvio Corso
02/12/2018	1.0	Glossario	Francesco Breve, Silvio Corso
13/12/2018	1.1	Modifica schema E/R e schema logico	Davide Bottiglieri, Silvio Corso, Vincenzo Sabato
06/01/2019	1.2	Modifica gestione dati persistenti	Silvio Corso
14/01/2019	1.3	Modifica gestione dati persistenti	Silvio Corso
20/01/2019	1.4	Modifica modello logico	Silvio Corso
21/01/2019	1.5	Modifica architettura proposta 3.1	Silvio Corso



## Team composition

Ruolo	Nome	Posizione	Contatti
<b>Top Manager</b>	Filomena Ferrucci	Rappresentante del cliente	<a href="mailto:f.ferrucci@unisa.it">f.ferrucci@unisa.it</a>
<b>Project Manager</b>	Francesco Vicidomini	Project Manager	<a href="mailto:f.vicidomini14@studenti.unisa.it">f.vicidomini14@studenti.unisa.it</a>
<b>Project Manager</b>	Federico Vitale	Project Manager	<a href="mailto:f.vitale40@studenti.unisa.it">f.vitale40@studenti.unisa.it</a>
<b>Team Member</b>	Alfonso Ruggiero		<a href="mailto:a.ruggiero114@studenti.unisa.it">a.ruggiero114@studenti.unisa.it</a>
<b>Team Member</b>	Davide Bottiglieri		<a href="mailto:d.bottiglieri4@studenti.unisa.it">d.bottiglieri4@studenti.unisa.it</a>
<b>Team Member</b>	Francesco Breve		<a href="mailto:f.breve@studenti.unisa.it">f.breve@studenti.unisa.it</a>
<b>Team Member</b>	Giuseppe Cavaliere		<a href="mailto:g.cavaliere10@studenti.unisa.it">g.cavaliere10@studenti.unisa.it</a>
<b>Team Member</b>	Paolo Cantarella		<a href="mailto:p.cantarella1@studenti.unisa.it">p.cantarella1@studenti.unisa.it</a>
<b>Team Member</b>	Rosaria Iorio		<a href="mailto:r.iorio11@studenti.unisa.it">r.iorio11@studenti.unisa.it</a>
<b>Team Member</b>	Silvio Corso		<a href="mailto:s.corso1@studenti.unisa.it">s.corso1@studenti.unisa.it</a>
<b>Team Member</b>	Vincenzo Sabato		<a href="mailto:v.sabato1@studenti.unisa.it">v.sabato1@studenti.unisa.it</a>



1. Introduzione.....	4
1.1 Obiettivi del sistema .....	4
1.2 Obiettivi di design.....	5
1.3 Definizioni, acronimi e abbreviazioni.....	7
1.4 Riferimenti .....	7
1.5 Panoramica .....	8
2 . Architettura del Sistema corrente.....	8
3. Architettura del sistema proposto .....	9
3.1 Panoramica .....	9
3.1.1 Vantaggi del Modello MVC.....	9
3.1.2 Svantaggi del Modello MVC .....	10
3.2 Decomposizione in sottosistemi .....	10
3.3 Mapping hardware/software.....	18
3.3.1 Componenti legacy and off the shelf.....	19
3.4 Gestione dati persistenti .....	19
3.4.1 Diagramma ER.....	19
3.4.2 Diagramma ER ristrutturato .....	20
3.4.3 Schema logico.....	21
3.4.4 Struttura delle tabelle.....	22
3.4.5 Controllo degli accessi.....	28
3.4.5.1 Tavola dei volumi .....	28
3.4.5.2 Tavola delle operazioni.....	29
3.4.5.3 Tavola degli accessi.....	30
3.5 Controllo della sicurezza.....	33
3.6 Controllo del flusso globale del sistema .....	35
3.7 Condizioni boundary.....	35
3.7.1 Avvio del sistema.....	35
3.7.2 Terminazione del sistema .....	36
3.7.3 Fallimento del sistema.....	37
4. Servizi dei sottosistemi.....	37



4.1 Gestione profilo utente.....	37
4.2 Gestione Forum.....	38
4.3 Gestione Coordinatori .....	40
4.4 Gestione Chat.....	41
5. Glossario .....	42

## 1. Introduzione

### 1.1 Obiettivi del sistema

Il Consiglio Didattico di Informatica dell'Università degli Studi di Salerno intende dare ai propri studenti la possibilità di migliorare e assumere nuove abilità utili per il mondo del lavoro, offrendo le esperienze Erasmus e Erasmus+ Traineeship in quanto esso è un criterio di valutazione delle università e va ad incrementare il piano di offerta formativa. Vuole quindi far incuriosire gli studenti a questo mondo agevolando anche i rapporti che intercorrono tra quest'ultimi e i responsabili di tale iniziativa; infatti molti studenti vengono spaventati dalla pesante burocrazia che bisogna affrontare per intraprendere tale esperienza e preferiscono rinunciare a questa importante opportunità.

Per svolgere l'Erasmus, infatti, c'è bisogno della stipulazione del Learning Agreement da parte dello studente, seguito da un coordinatore. L'obiettivo del sistema è rivolto sia allo studente che al coordinatore, in quanto:

- Si vuole agevolare lo studente ad intraprendere questo tipo di esperienza, velocizzando la ricerca di informazioni e la visualizzazione di avvisi tramite un'apposita bacheca online, dandogli anche la possibilità di consultare un forum sul quale può cercare informazioni o rivolgere delle domande.
- Si vuole agevolare il lavoro del coordinatore, mettendogli a disposizione gli strumenti per tenere traccia degli studenti partiti e tornati ma anche dei vari documenti legati al singolo coordinato per velocizzare i tempi di recupero.

## 1.2 Obiettivi di design

Rank	ID Design Goal	Descrizione Design Goal	Categoria	Origine	Trade off
1	DG_1 Tempo di risposta	Il sistema deve garantire tempi di risposta brevi per ogni funzionalità, complessivamente inferiori ai 5 secondi considerando una connessione media di 7 Mbps.	Performance	RNF_4	Se il software non rispetta i requisiti di tempo di risposta e di throughput, è necessario utilizzare più memoria per velocizzare il sistema. Altrimenti, se il software non rispetta i requisiti di memoria, questa può essere compressa a discapito della velocità
2	DG_2 Memoria	Si utilizzerà il database relazionale MySQL, per gestire la persistenza dei dati.	Performance	RNF_4	
3	DG_3 Throughput	Il sistema sarà capace di gestire contemporaneamente e diversi utenti. In caso di elevato numero di accessi concorrenti, il sistema mostrerà, all'utente che tenta di autenticarsi, un messaggio che li inviti ad accedere in un altro momento.	Performance	RNF_4	
4	DG_4 Robustezza	Verrà visualizzato un messaggio di errore che avvertirà l'utente in caso di dati inseriti non validi o mancanti.	Dependability	RNF_3	Un budget limitato non garantisce un livello di robustezza molto elevato
Rank	ID Design Goal	Descrizione Design Goal	Categoria	Origine	Trade off
5	DG_5 Sicurezza	Ogni utente è dotato di e-mail e password	Dependability	RNF_3	

		con cui poter effettuare l'accesso al sistema e utilizzare solo le funzionalità associate alla propria categoria.			
6	DG_6 Costi	Non avere costi eccessivi per la costruzione dell'architettura.	Cost	Business case	
7	DG_7 Tracciabilità dei requisiti	Utilizzo di una matrice di tracciabilità.	Maintenance	Matrice di tracciabilità	
8	DG_8 Usabilità	Il sistema sarà facilmente utilizzabile, intuitivo e user-friendly.	Dependability	RNF_2	
9	DG_9 Affidabilità	Il sistema garantirà l'affidabilità, intesa come corretta gestione delle funzionalità. Il 75% delle componenti del sistema dovrà essere testato e funzionante.	Dependability	RNF_3	
10	DG_10 Tolleranza ai guasti	Si eviterà il fallimento di operazioni, la dispersione e la perdita di dati.	Dependability	RNF_4	
11	DG_11 Disponibilità	Il sistema dovrà essere disponibile 24 ore su 24, 365 giorni all'anno.	Dependability	RNF_4	

12	DG_12 Manutenibilità	Il sistema dovrà essere facilmente manutenibile.	Maintenance	RNF_5	
13	DG_13 Estendibilità	Si potranno inserire nuove funzionalità al sistema. In futuro il sistema potrà diventare un'applicazione mobile disponibile per dispositivi IOS e Android.	Maintenance	RNF_5	

### 1.3 Definizioni, acronimi e abbreviazioni

- **SE:** nome del sistema in sviluppo, l'acronimo sta per "SharErasmus".
- **User-friendly:** aggettivo utilizzato per definire un software di facile utilizzo anche per persone non esperte nell'utilizzo del computer.

### 1.4 Riferimenti

- Kathy Schwalbe, "Information Technology Project Management", International Edition 7E, Cengage Learning, 2014;
- Bernd Bruegge, Allen H. Dutoit, "Object-Oriented Software Engineering Using UML, Patterns and Java", Third Ed., Pearson, 2010;
- Sommerville, "Software Engineering", Addison Wesley;
- PMBOK® Guide and Software Extension to the PMBOK® Guide, Fifth Ed., Project Management Institute, 2013
- Documentazione di Progetto:
  - Statement of Work;
  - SE\_RADDRAFT\_Vers1.0.docx



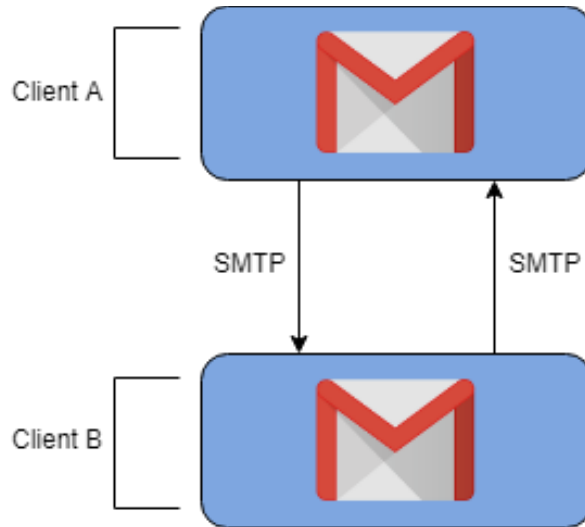
## 1.5 Panoramica

Il documento si compone di una prima parte in cui vengono introdotti gli obiettivi di design. Verrà poi brevemente spiegata l'architettura del sistema corrente e l'architettura del sistema proposto. Verranno brevemente illustrate le parti di cui si compone il documento:

- Nel capitolo 2 viene mostrata l'architettura del sistema corrente;
- Nel capitolo 3 viene mostrata l'architettura del sistema proposto fornendo una panoramica, e mostrando in dettaglio:
- Decomposizione del sistema in sottosistemi di funzionalità e operazioni correlate da assegnare a vari membri del team in fase di sviluppo;
- Mapping Hardware/Software, riguardante la scelta della configurazione hardware del sistema, le componenti “off-the-shelf”, la comunicazione tra nodi, incapsulamento dei servizi di un sottosistema;
- Gestione dei dati persistenti descrive i dati persistenti memorizzati dal sistema e l'infrastruttura di gestione richiesta per memorizzarli;
- Controllo degli accessi e della sicurezza descrive, tramite una matrice degli accessi, le operazioni effettuabili e le informazioni consultabili da ogni tipologia di utente e come questi si autenticano al sistema;
- Controllo del flusso globale descrive quali operazioni eseguire ed in che ordine, per garantire il corretto flusso di controllo del sistema;
- Condizioni boundary che includono oltre l'avvio e lo spegnimento del sistema, la gestione dei fallimenti dovuti all'invecchiamento, interruzione di corrente o anche a errori di progettazione.
- Nel capitolo 4 vengono mostrati i servizi forniti da ogni sottosistema ed individua i loro confini.
- Nel capitolo 5 viene fornito il glossario dei termini utilizzati nel documento con le relative definizioni.

## 2 . Architettura del Sistema corrente

Non esiste alcuna architettura software che soddisfi le problematiche dell'Erasmus al momento in quanto il sistema corrente è gestito principalmente attraverso uno scambio di mail tra utenti e coordinatori. Il sistema corrente presenta questa architettura:



### 3. Architettura del sistema proposto

Il sistema che si vuole proporre deriva da un progetto di “Greenfield Engineering”.

#### **3.1 Panoramica**

L'architettura scelta per questo sistema è il Model-View-Controller (MVC). Essa si struttura in tre componenti, ognuno con un compito diverso all'interno del sistema:

- il model fornisce i metodi per accedere ai dati utili all'applicazione;
- il view visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti;
- il controller riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti.

##### ***3.1.1 Vantaggi del Modello MVC***

- MVC supporta lo sviluppo rapido e parallelo. Se un modello MVC viene utilizzato per sviluppare una particolare applicazione Web, è possibile che un programmatore possa lavorare su una view mentre un altro su un controller per creare la logica di business dell'applicazione web. Quindi, in questo modo, l'applicazione sviluppata utilizzando il modello MVC può essere completata tre volte più velocemente rispetto alle applicazioni sviluppate utilizzando altri modelli di sviluppo;
- Nel modello MVC, è possibile creare più view per un model. Inoltre, grazie a questo modello, la duplicazione del codice è molto limitata perché vengono separati i dati e la logica di business dalle interfacce grafiche;
- L'architettura MVC si integra perfettamente con JavaScript. Ciò significa che le applicazioni MVC possono funzionare con file PDF, browser specifici e widget desktop. MVC supporta

anche la tecnica asincrona, che aiuta gli sviluppatori a sviluppare un'applicazione che ha tempi di caricamento molto veloci;

- Per qualsiasi applicazione Web, l'interfaccia utente tende a cambiare molto velocemente. Vengono apportati di continuo modifiche alla propria applicazione Web, come cambiamenti nei colori, caratteri, layout dello schermo e aggiunta di nuovo supporto per dispositivi mobili (cellulari o tablet). L'aggiunta di un nuovo tipo di view è molto semplice con il modello MVC poiché la parte Model non dipende dalla parte View. Pertanto, eventuali modifiche nel Model non influenzeranno l'intera architettura;
- Il pattern MVC restituisce i dati senza applicare alcuna formattazione. Quindi, gli stessi componenti possono essere utilizzati e richiamati per l'uso con qualsiasi interfaccia. Ad esempio, qualsiasi tipo di dati può essere formattato con HTML, ma potrebbe anche essere formattato con Macromedia Flash o Dream viewer.

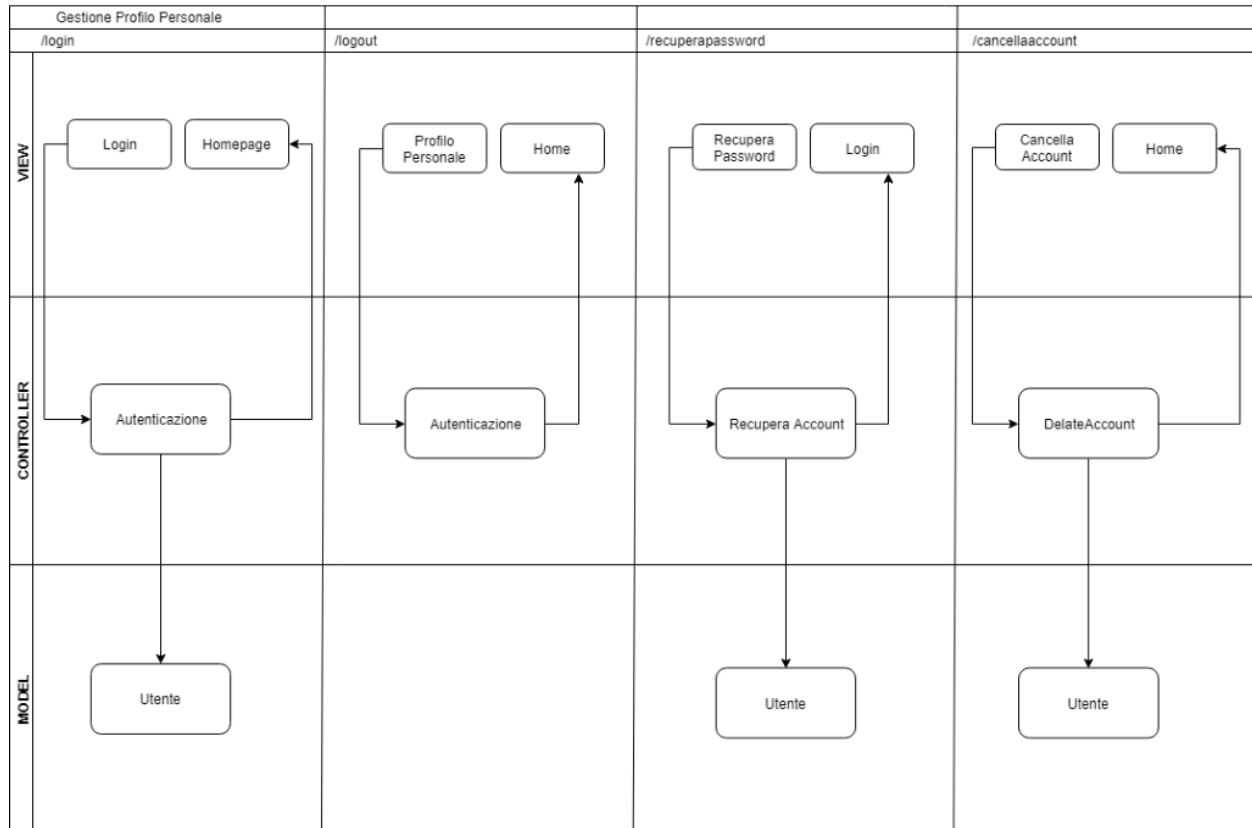
### ***3.1.2 Svantaggi del Modello MVC***

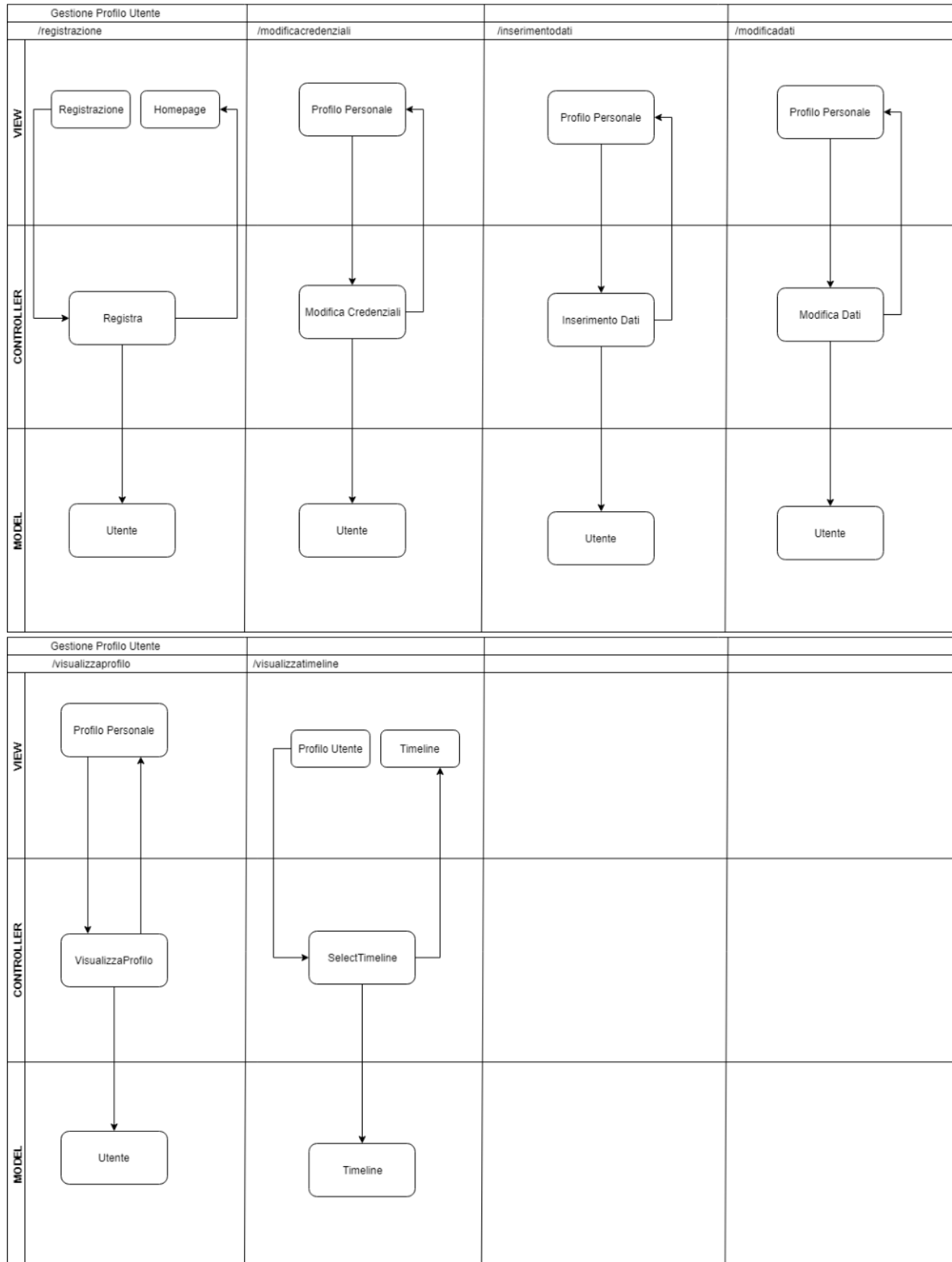
- Modello di design complesso. Ha bisogno di una buona conoscenza del flusso di controllo tra view, logica di business e controller. Altrimenti sarà difficile eseguire il debug;
- Se diversi team lavorano su classi UI e Model, allora devono avere lo stesso ritmo di sviluppo;
- Il processo di sviluppo isolato per la UI, la logica di business e i controller può portare a ritardi nello sviluppo dei rispettivi moduli.

## **3.2 Decomposizione in sottosistemi**

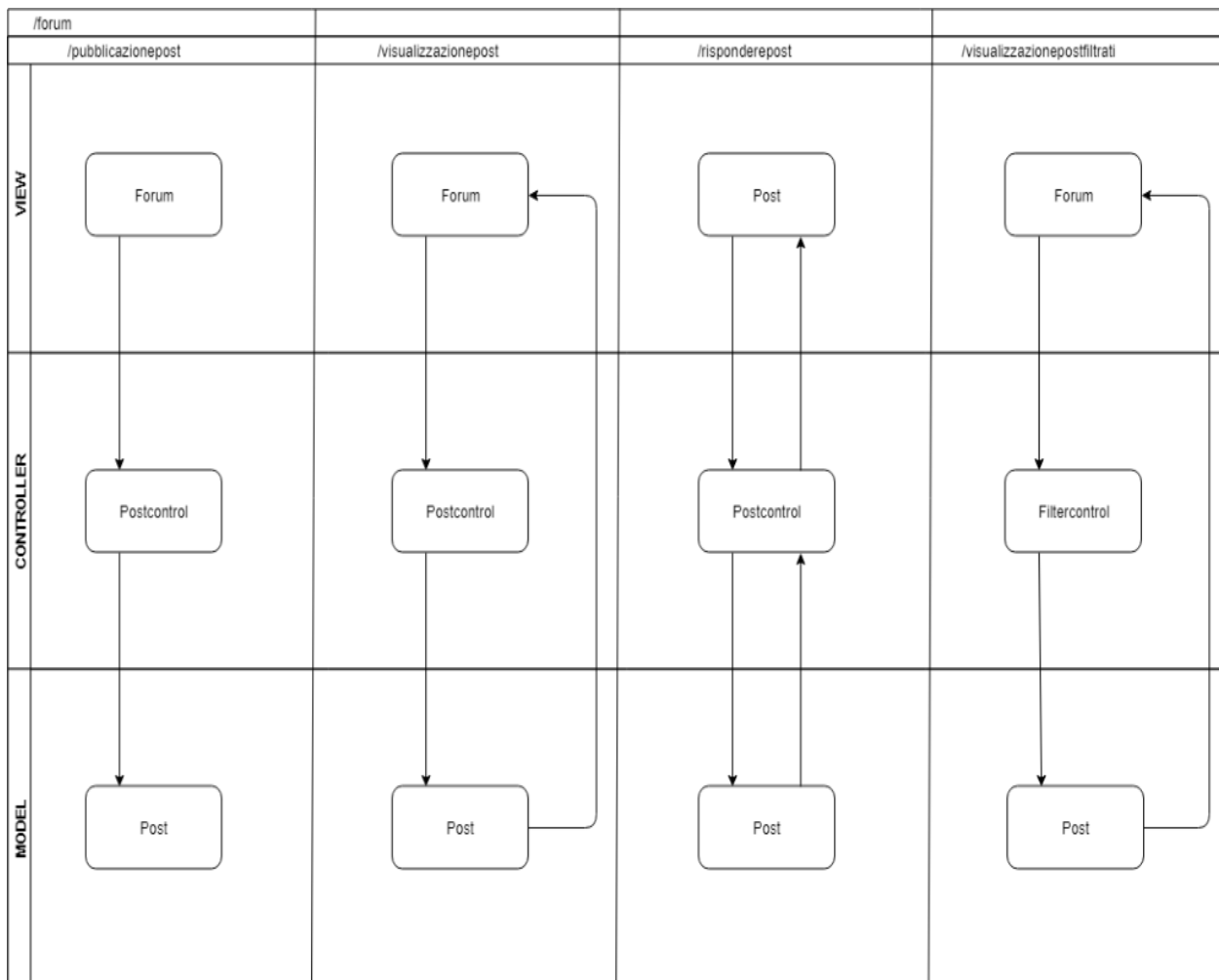
Il sistema è diviso in 4 gestioni principali: **Gestione profilo personale**, **Gestione forum**, **Gestione coordinatori** e **Gestione chat**.

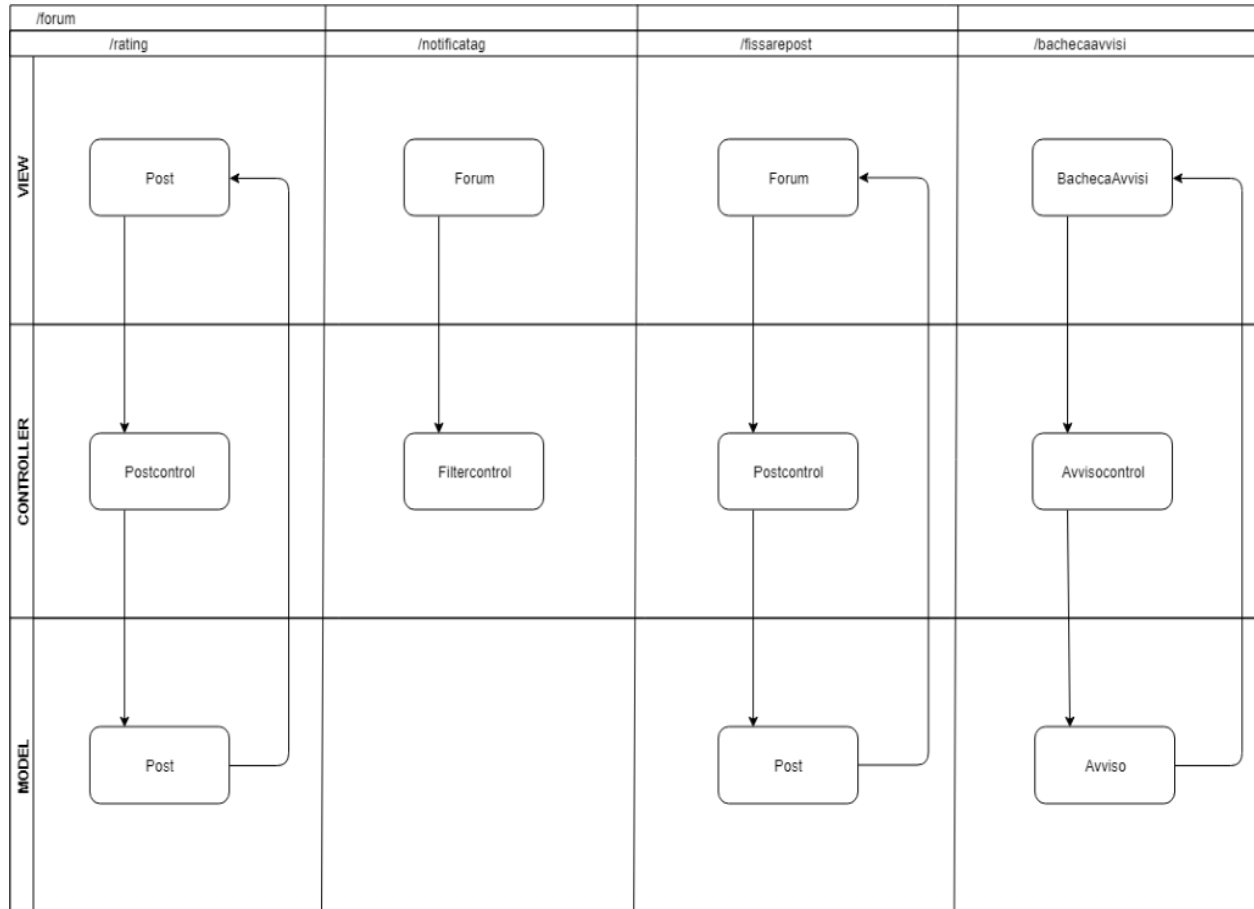
La **Gestione del profilo utente** si occupa dell'autenticazione dell'utente, della sua registrazione, del recupero password e della modifica del proprio profilo.



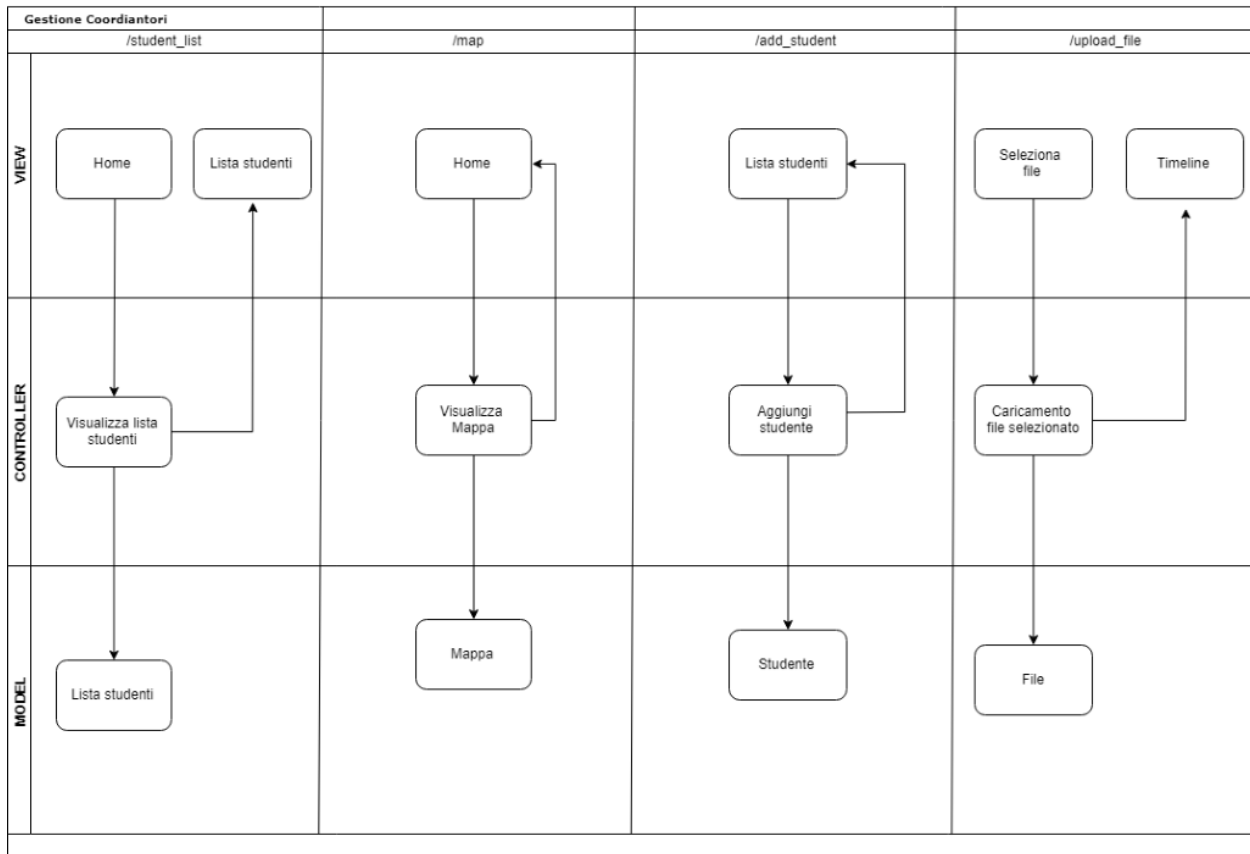


La **Gestione del forum** si occupa della visualizzazione, pubblicazione, messa in evidenza e interazione con i post come pubblicazione di commenti e sistema di rating. Inoltre, si occupa anche del sistema di filtri gestito tramite tag. Sarà inoltre presente una bacheca avvisi che non conterrà post bensì annunci per dare comunicazioni o segnalare scadenze in modo rapido ed efficace.

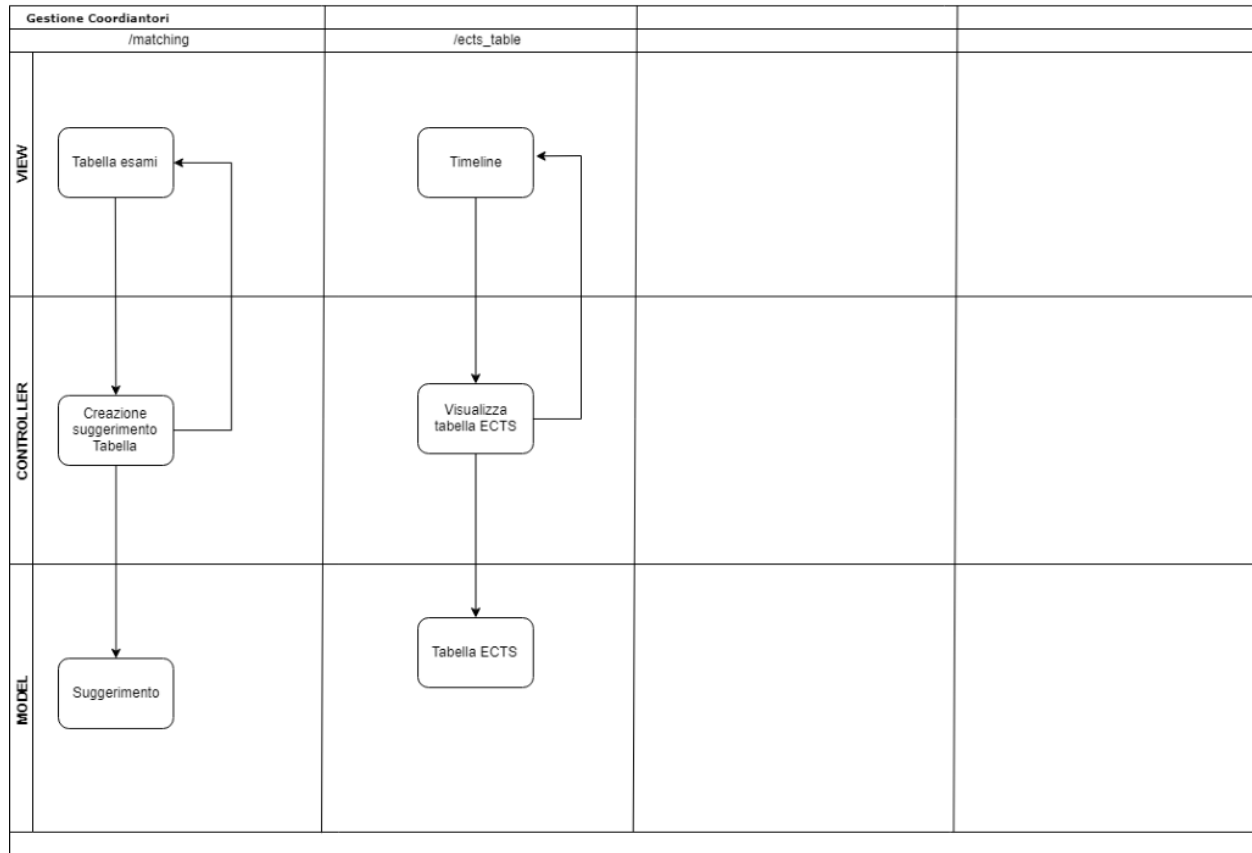




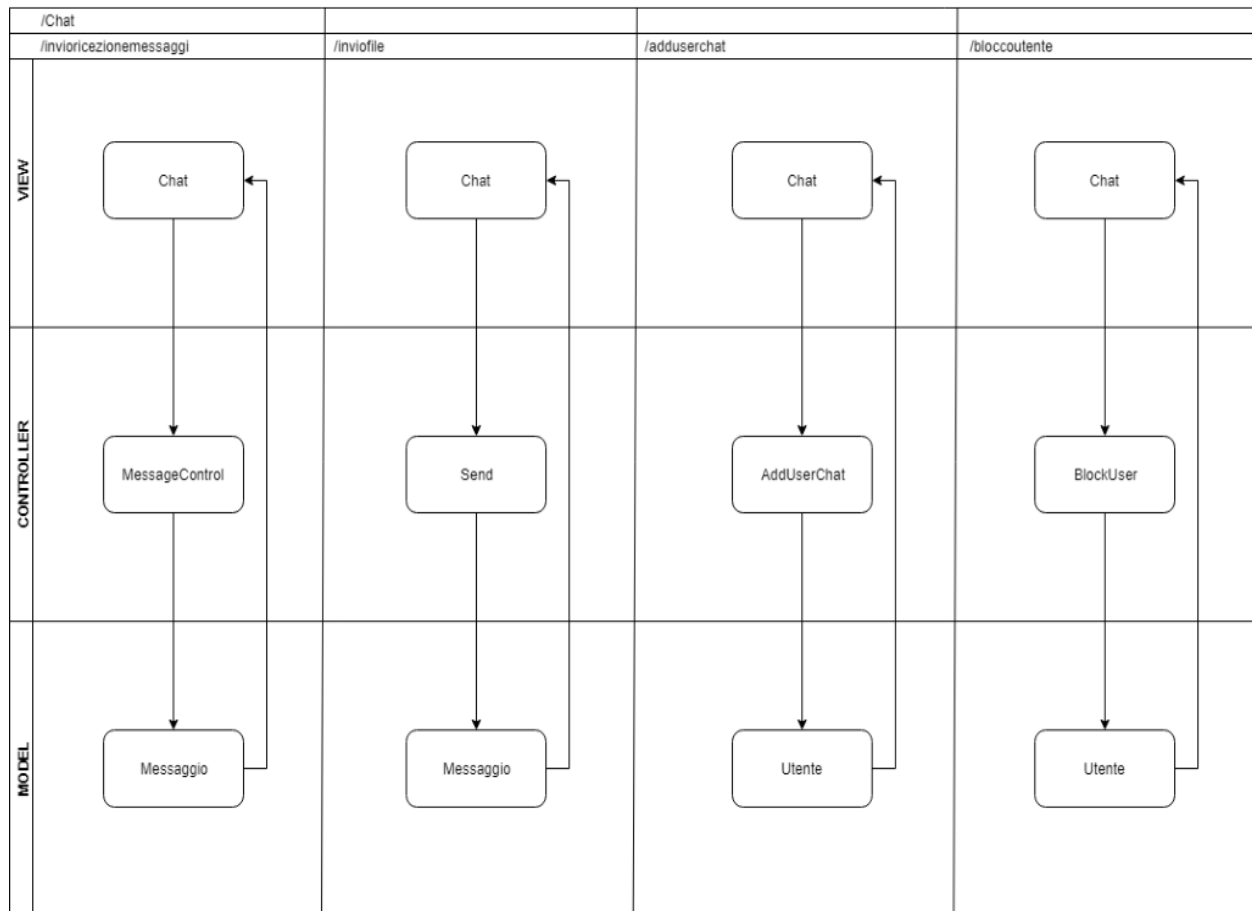
La **Gestione Coordinatori** è uno strumento utile ai coordinatori per tenere traccia dei documenti e degli esami conseguiti dagli studenti Erasmus che essi coordinano.



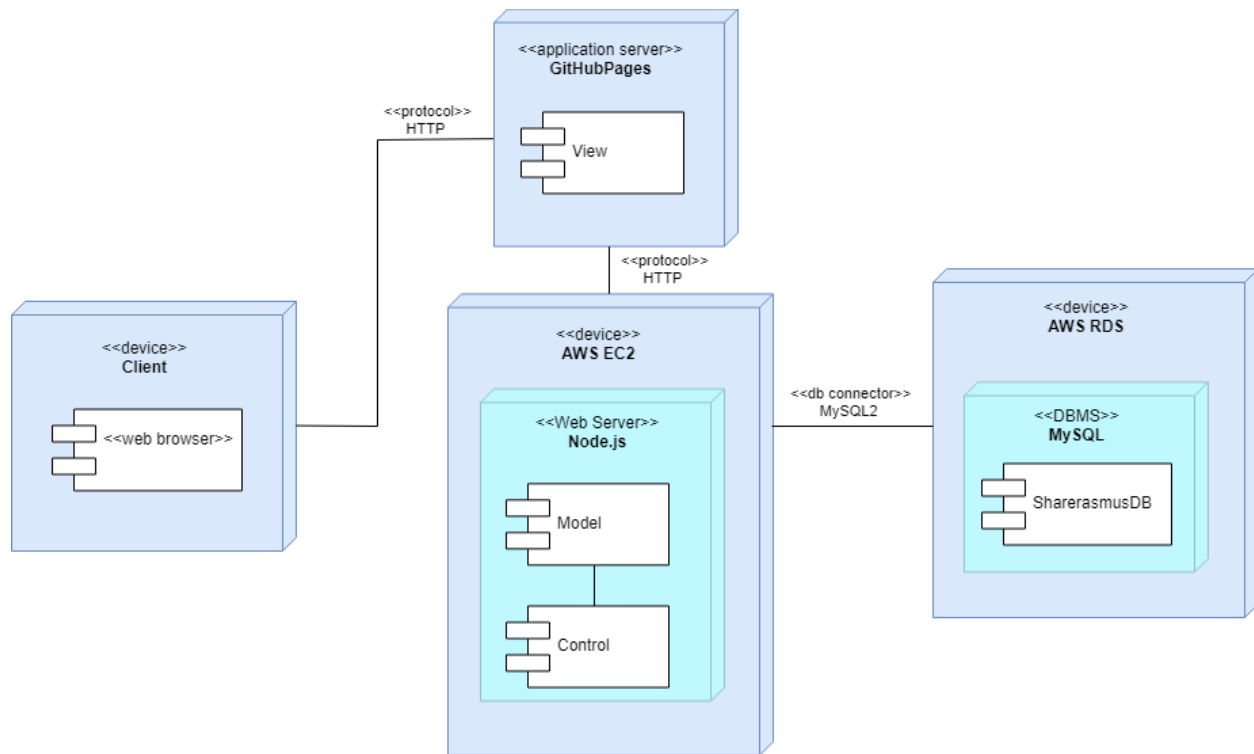




La **Gestione Chat** si occupa della messaggistica tra gli utenti, dell'invio di file tra di loro, della creazione di chat di gruppo e della possibilità di bloccare un utente.



### 3.3 Mapping hardware/software



Il sistema si presenta con un'architettura di tipo client/server divisa in tre strati logici (layer) e quattro macchine fisiche (tier), in cui un server espone i servizi ad uno o più client. Il primo layer, il client sarà eseguito da un qualsiasi browser web che supporti HTML5, permetterà all'utente di interagire con le view. Il secondo layer è diviso in due sottolivelli, il web server che conterrà model e control e il site hosting che si occuperà di mantenere solo le view. Il primo sottolivello da Amazon Web Service Elastic Compute Cloud (AWS EC2), un servizio di Amazon che permette di istanziare macchine virtuali. Node.js gestisce la logica applicativa (model e control). Il secondo sottolivello invece consiste nel servizio di site hosting, GitHubPages per conservare tutte le view e minimizzare così i costi di hosting, in quanto GitHubPages è un servizio di hosting gratuito che facilita il deploy delle pagine web, poiché non richiede il passaggio per il protocollo FTP (File Transfer Protocol) ma basta semplicemente effettuare un commit sul repository GitHub in cui sono situate le view e sarà così possibile modificarne o aggiungerne di nuove. Il Client e Server comunicheranno tramite il protocollo HTTP, un protocollo di trasferimento di ipertesti che consente a due macchine di tipo client e server di interagire tramite request e response. Lo stesso tipo di comunicazione avverrà tra AWS EC2 e GitHubPages. Il terzo layer invece, si occuperà della gestione dei dati persistenti tramite il DBMS MySQL. Per mantenere il database accessibile da remoto useremo un'altro servizio di Amazon, AWS RDS. La comunicazione tra Server e Database avviene tramite MySQL2 un connector che permette interazioni tra Node.js e MySQL concentrandosi sulle performance.

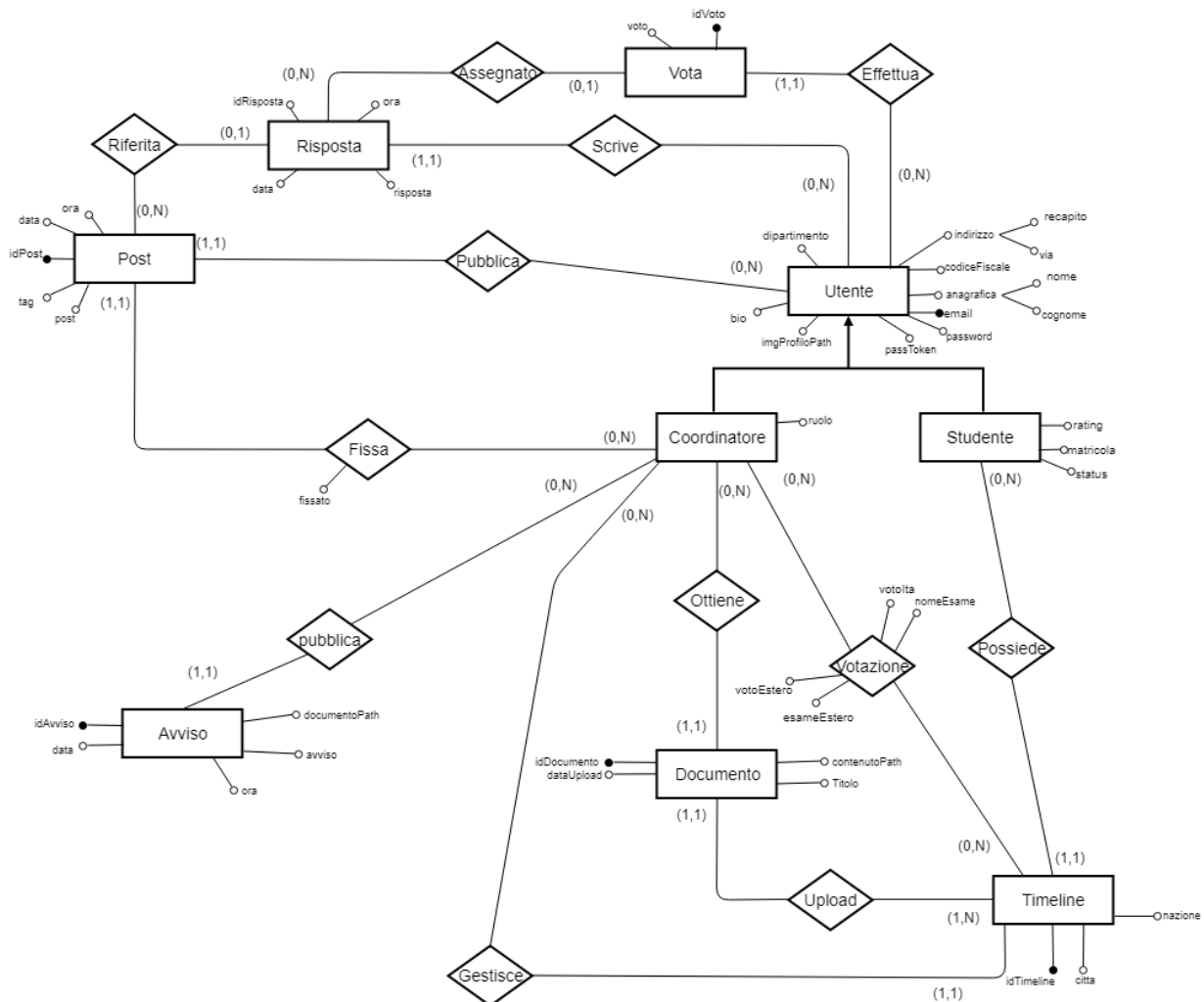
### 3.3.1 Componenti legacy and off the shelf

Dato che il sistema nasce ex novo non presenta componenti legacy.

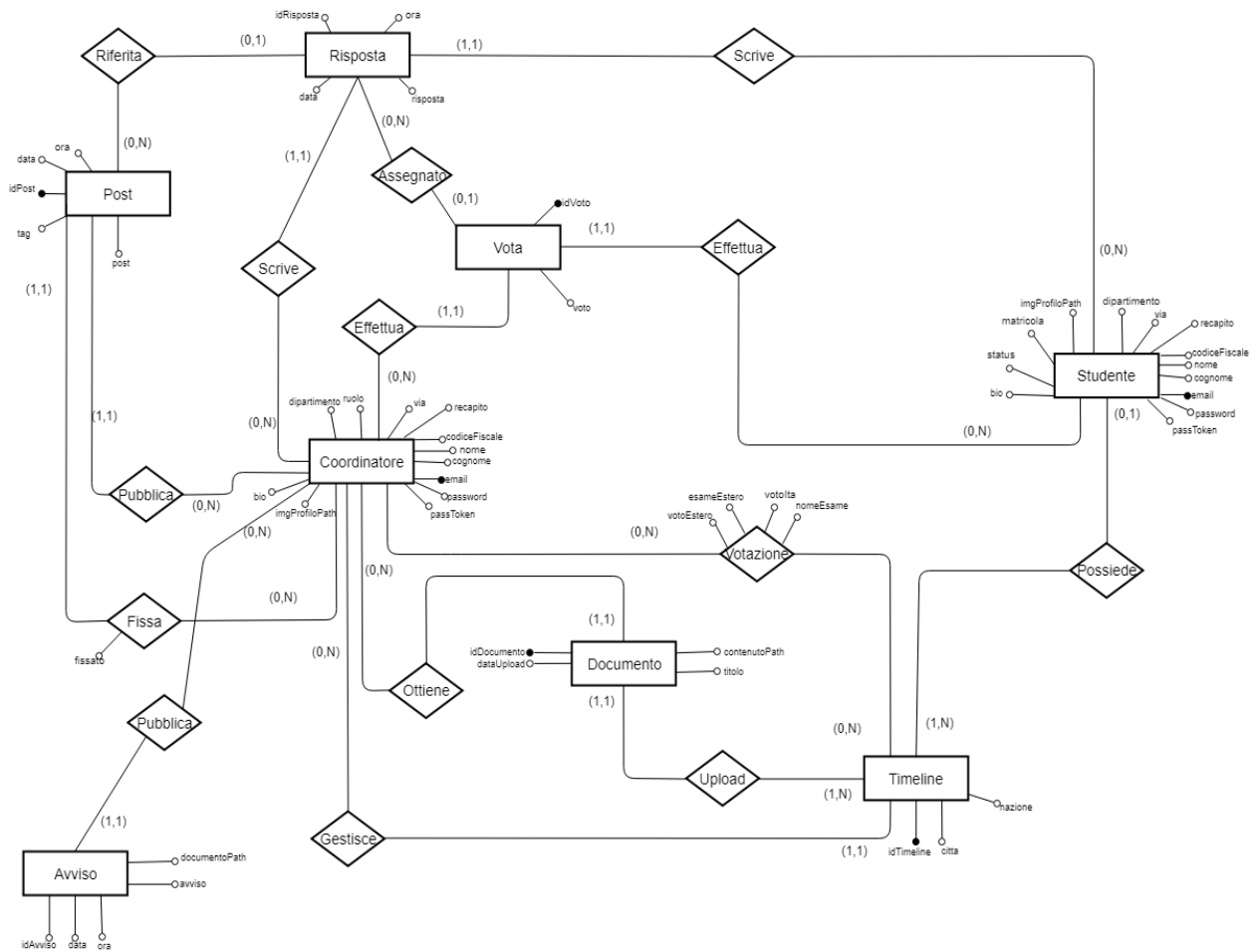
Per la gestione chat ci si affiderà a un componente off the shelf, “Firebase”, la quale è un'applicazione mobile e web sviluppata dall'omonima azienda, acquistata successivamente da Google; quest'ultima presenta varie funzionalità di cui noi utilizzeremo “Firebase Cloud Messaging” e “Firebase Realtime Database”.

## 3.4 Gestione dati persistenti

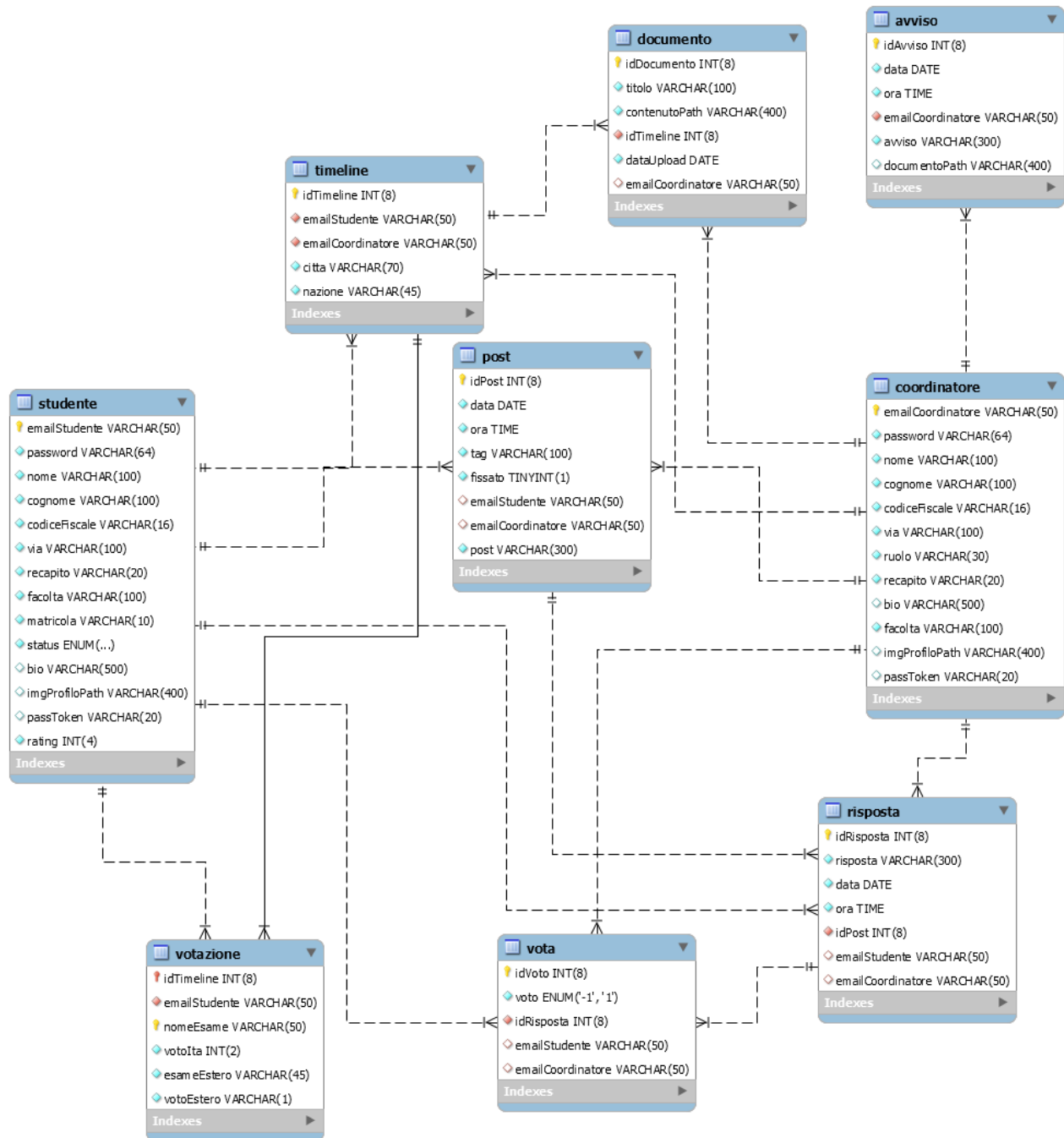
### 3.4.1 Diagramma ER



### 3.4.2 Diagramma ER ristrutturato



### 3.4.3 Schema logico



### 3.4.4 Struttura delle tabelle

Di seguito sono riportate tutte le tabelle che vanno a formare il nostro database per la gestione di tutte le informazioni del sistema. In ogni tabella è indicato: il nome, il compito e tutti gli attributi ad essa associati, con relativi vincoli e tipo.

Coordinatore		
Campo	Vincoli	Tipo
email	Lunghezza massima: 50 caratteri; Primary key	Varchar
password	Lunghezza massima:16 caratteri; Not null	Varchar
nome	Lunghezza massima: 100 caratteri; Not null	Varchar
cognome	Lunghezza massima: 100 caratteri; Not null	Varchar
codiceFiscale	Lunghezza massima: 16 caratteri; Not null	Varchar
via	Lunghezza massima: 100 caratteri; Not null	Varchar
recapito	Lunghezza massima: 20 caratteri; Not null	Varchar
ruolo	Lunghezza massima: 30 caratteri; Not null	Varchar
imgProfiloPath	Lunghezza massima: 400 caratteri;	Varchar



	Not null	
passwordToken	Lunghezza massima: 20 caratteri; Not null	Varchar
facolta	Lunghezza massima: 100 caratteri; Not null	Varchar

Studente		
Campo	Vincoli	Tipo
email	Lunghezza massima: 50 caratteri; Primary key	Varchar
password	Lunghezza massima:16 caratteri; Not null	Varchar
nome	Lunghezza massima: 100 caratteri; Not null	Varchar
cognome	Lunghezza massima: 100 caratteri; Not null	Varchar
codiceFiscale	Lunghezza massima: 16 caratteri; Not null	Varchar
via	Lunghezza massima: 100 caratteri; Not null	Varchar
recapito	Lunghezza massima: 20 caratteri; Not null	Varchar
facolta	Lunghezza massima: 100 caratteri; Not null	Varchar
matricola	Lunghezza massima: 10 caratteri;	Varchar

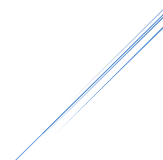


	Not null	
status	Lunghezza massima: Può assumere solo valori Normale, Tornato, Partito; Not null	Enum
imgProfiloPath	Lunghezza massima: 400 caratteri;	Varchar
passwordToken	Lunghezza massima: 20 caratteri;	Varchar
rating	Lunghezza massima: 4 cifre; Not null	Integer

Post		
Campo	Vincoli	Tipo
idPost	Primary key	Integer
data	Not null	Date
ora	Precisione orario nel formato hh-mm-ss; Not null	Time
tag	Lunghezza massima: 100 caratteri; Not null	Varchar
fissato	Può assumere solo i valori: 0, 1; Not null	TinyInt
emailStudente	Lunghezza massima: 50 caratteri; Foreign key	Varchar
emailCoordinatore	Lunghezza massima: 50 caratteri; Foreign key	Varchar

Vota		
Campo	Vincoli	Tipo
idVoto	Primary key	Integer
voto	Puo' assumere i valori (-1,1) Not null	Enum
idRisposta	Foreign key; Not null	Integer
emailStudente	Lunghezza massima: 50 caratteri; Foreign key	Varchar
emailCoordinatore	Lunghezza massima: 50 caratteri; Foreign key	Varchar

Risposta		
Campo	Vincoli	Tipo
idRisposta	Lunghezza massima: 8 cifre; Primary key	Integer
data	Not null	Date
ora	Precisione orario nel formato hh-mm-ss; Not null	Time
risposta	Lunghezza massima: 300 caratteri; Not null	Varchar
idPost	Lunghezza massima: 8 cifre; Foreign key; Not null	Integer
emailStudente	Lunghezza massima: 50 caratteri; Foreign key;	Varchar
emailCoordinatore	Lunghezza massima: 50 caratteri; Foreign key;	Varchar



Timeline		
Campo	Vincoli	Tipo
idTimeline	Lunghezza massima: 8 cifre; Primary key	Integer
emailStudente	Lunghezza massima: 50 caratteri; Foreign key; Not null	Varchar
citta	Lunghezza massima: 70 caratteri; Not null	Varchar
nazione	Lunghezza massima: 45 caratteri; Not null	Varchar

Votazione		
Campo	Vincoli	Tipo
idTimeline	Lunghezza massima: 8 cifre; Primary key, Foreign key	Integer
emailStudente	Lunghezza massima: 50 caratteri; Foreign key	Varchar
nomeEsame	Lunghezza massima: 50 caratteri; Not null Primary key	Varchar
votolta	Lunghezza massima: 2 cifre; Not null	Integer
esameEstero	Lunghezza massima: 50 caratteri; Not null	Varchar
votoEstero	Lunghezza massima: 1 carattere;	Varchar

	Not null	
--	----------	--

Documento		
Campo	Vincoli	Tipo
idDocumento	Lunghezza massima: 8 cifre; Primary key	Integer
tipo	Lunghezza massima: 5 caratteri; Not null	Varchar
contenutoPath	Lunghezza massima: 400 caratteri; Not null	Varchar
titolo	Lunghezza massima: 100 caratteri; Not null	Varchar
idTimeline	Lunghezza massima: 8 cifre; Foreign key; Not null	Integer
emailCoordinatore	Lunghezza massima: 50 caratteri; Foreign key;	Varchar
dataUpload	Not null;	Date

Avviso		
Campo	Vincoli	Tipo
idAvviso	Lunghezza massima: 8 cifre; Primary key	Integer
avviso	Lunghezza massima: 300 caratteri; Not null	Varchar
data	Not null	Date
ora	Precisione orario nel formato hh-mm-ss;	Time

	Not null	
documentoPath	Lunghezza massima: 400 caratteri; Not null	Varchar
emailCoordinatore	Lunghezza massima: 50 caratteri; Foreign key;	Varchar

### 3.4.5 Controllo degli accessi

#### 3.4.5.1 Tavola dei volumi

Nel 2015 sono stati 33.977 gli studenti italiani interessati a compiere un'esperienza Erasmus di cui 7.666 interessati ad un Erasmus per tirocinio. SharErasmus nasce nell'Università di studi di Salerno e conta di partire quindi con un bacino di utenza molto ristretto, circa 3000 studenti. Si auspica in oltre la presenza di oltre 150 coordinatori, tra professori italiani ed esteri. Ogni studente può avere da 0 a 2 timeline (una per ogni coordinatore responsabile dello studente), tuttavia supponendo che non tutti i coordinatori esteri si servano del sistema si prevede una media di circa 1.5 timeline per studente, salendo a 4500 timeline annue. Ogni coordinatore può caricare dei documenti su ogni timeline, si ci aspetta quindi circa 20.250 documenti di vario tipo caricati ogni anno, supponendo che ogni documento venga caricato su due timeline differenti, una per ogni coordinatore la funzione di upload documento verrà usata 40.500 volte all'anno. SharErasmus implementerà anche un forum in cui ci si aspetta 1 post ogni giorno per ogni utente registrato, circa 1.314.000 post di cui l'80% viene da studenti (1.051.200) e il restante 20% da coordinatori (262.800). Ogni post potrà ricevere una o più risposte, si stima circa 3.942.000 risposte annue mentre per i coordinatori supponiamo forniranno risposta al 25% dei post totale quindi 328.500 risposte da parte di un docente. SharErasmus fornisce anche la possibilità di votare ogni post o risposta, considerando che gli utenti attivi votino da 1 a 3 post al giorno, quindi circa 5.913.000 voti registrati ogni anno di cui il 90% (5.321.700) da parte di studenti e, il restante 10% (591.320) da parte di coordinatori. Ogni coordinatore può mettere in evidenza un post per fornirgli massima visibilità, questa funzionalità dovrebbe essere applicata sul 15% dei post di ogni coordinatore quindi 39.450 volte.

Concetto	Costrutto	Volume
Studente	E	3000
Risponde (studente-post)	R	3.942.000
Risponde (coordinatore-post)	R	328.500
Pubblica (studente-post)	R	1.051.200
Pubblica (coordinatore-post)	R	262.800

Coordinatore	E	150
Fissa	R	39.450
Ottiene	R	18.255(90% dei documenti totali)
Timeline	E	4500
Possiede	R	4500
Votazione	R	405.000(ogni coordinatore effettua 3 votazione per ogni timeline ed è assegnato al massimo al 20% delle timeline totali 150x3x900)
Documento	E	20.250
Upload	R	40.500
Post	E	1.314.000
Assegna	R	3.022.220
Effettua (studente-voto)	R	5.321.700
Effettua (studente-coordinatore)	R	591.320
Voto	E	5.913.000
Gestisce	R	4500(Ogni coordinatore gestisce 30 timeline 150x30)

#### 3.4.5.2 Tavola delle operazioni

Operazione	Tipo	Frequenza
Registrazione	Interattiva	4000/anno
Login	Interattiva	73000/anno
Inserisci post	Interattiva	37000/anno
Rispondi	Interattiva	73000/anno
Voto utente	Interattiva	73000/anno

Raggruppa per tag	Interattiva	12200/anno
Crea Timeline	Interattiva	100/anno
Carica Documento	Interattiva	300/anno
Voto esame	Interattiva	600/anno
Visualizza Profilo	Interattiva	36500/anno

### 3.4.5.3 Tavola degli accessi

#### Login

Concetto	Costrutto	Accessi	Tipo Accessi
Studente/ Coordinatore	E	1	L

**Totale accessi** =  $1 \times 73000 = 73000$  accessi / anno

#### Registrazione

Concetto	Costrutto	Accessi	Tipo Accessi
Studente	E	1	L
Studente	E	1	S

**Totale accessi** =  $1 + (1 \times 2) = 3 \times 4000 = 12000$  accessi / anno

#### Inserisci post

Concetto	Costrutto	Accessi	Tipo Accessi
Studente/ Coordinatore	E	1	L
Pubblica	R	1	S
Post	E	1	S

**Totale accessi** =  $1 + (1 + 1) \times 2 = 4 \times 37000 = 148000$  accessi / anno



*Rispondi*

Concetto	Costrutto	Accessi	Tipo Accessi
Studente/ Coordinatore	E	1	L
Risponde	R	1	S
Post	E	1	S

**Totale accessi** =  $1 + (1+1) \times 2 = 4 \times 73000 = 292000$  accessi / anno

*Voto utente*

Concetto	Costrutto	Accessi	Tipo Accessi
Studente/ Coordinatore	E	1	L
Effettua	R	1	S
Voto	E	1	S
Assegna	R	1	S

**Totale accessi** =  $1 + (1+1+1) \times 2 = 7 \times 73000 = 511000$  accessi / anno

*Raggruppa per tag*

Concetto	Costrutto	Accessi	Tipo Accessi
Post	E	1	L

**Totale accessi** =  $1 \times 12200 = 12200$  accessi / anno



*Crea timeline*

Concetto	Costrutto	Accessi	Tipo Accessi
Studente	E	1	L
Possiede	R	1	S
Coordinatore	E	1	L
Gestisce	R	1	S
Timeline	E	1	S

**Totale accessi** =  $1+1+(1+1+1) \times 2 = 8 \times 100 = 800$  accessi / anno

*Carica documento*

Concetto	Costrutto	Accessi	Tipo Accessi
Documento	E	1	L
Upload	R	1	S

**Totale accessi** =  $1+(1 \times 2) = 3 \times 300 = 900$  accessi / anno

*Voto esame*

Concetto	Costrutto	Accessi	Tipo Accessi
Coordinatore	E	1	L
Timeline	E	1	L
Votazione	R	1	S

**Totale accessi** =  $1+1+(1 \times 2) = 4 \times 600 = 2400$  accessi / anno

*Visualizza profilo*

Concetto	Costrutto	Accessi	Tipo Accessi
Studente	E	1	L

**Totale accessi** = 1 x 36500 = 36500 accessi / anno

### 3.5 Controllo della sicurezza

Il software SharErasmus permette l'accesso a 3 tipi di utenti diversi: coordinatori, studenti e utenti non registrati. Per poter effettuare l'accesso alla piattaforma come studente o coordinatore bisognerà inserire le credenziali utilizzate nella registrazione (email e password) altrimenti si potrà utilizzarla come utente registrato ma con diverse restrizioni sulle proprie azioni.

Attori	Gestione profilo utente	Gestione forum	Gestione coordinatore	Gestione chat
Coordinatore	<ul style="list-style-type: none"> <li>Visualizzare il proprio profilo</li> <li>Visualizzare profilo di un altro utente</li> <li>Modificare biografia</li> <li>Modificare credenziali</li> <li>Modificare immagine del profilo</li> <li>Recuperare password</li> </ul>	<ul style="list-style-type: none"> <li>Visualizzare post</li> <li>Visualizzare dettagli post</li> <li>Rispondere ad un post</li> <li>Filtrare post per tag</li> <li>Pubblicare un post</li> <li>Pubblicare un annuncio</li> <li>Votare un post</li> <li>Mettere in evidenza un post</li> </ul>	<ul style="list-style-type: none"> <li>Visualizzare lista studenti coordinati partiti/non partiti/tornati</li> <li>Aggiungere uno studente alla lista</li> <li>Caricare documenti nella timeline di uno studente</li> <li>Convalidare voti degli esami di uno studente</li> </ul>	<ul style="list-style-type: none"> <li>Visualizzare utenti online/offline</li> <li>Iniziare una conversazione</li> <li>Ricevere messaggi</li> <li>Creare chat di gruppo</li> <li>Partecipare ad una chat di gruppo</li> <li>Bloccare/Sbloccare utenti</li> </ul>
Studente	<ul style="list-style-type: none"> <li>Visualizzare il proprio profilo</li> </ul>	<ul style="list-style-type: none"> <li>Visualizzare post</li> <li>Visualizzare dettagli post</li> </ul>	<ul style="list-style-type: none"> <li>Visualizzare la propria timeline</li> </ul>	<ul style="list-style-type: none"> <li>Visualizzare utenti online/offline</li> </ul>

	<ul style="list-style-type: none"> <li>• Visualizzare profilo di un altro utente</li> <li>• Modificare biografia</li> <li>• Modificare credenziali</li> <li>• Modificare immagine del profilo</li> <li>• Recuperare password</li> </ul>	<ul style="list-style-type: none"> <li>• Rispondere ad un post</li> <li>• Filtrare post per tag</li> <li>• Pubblicare un post</li> <li>• Votare un post</li> </ul>		<ul style="list-style-type: none"> <li>• Iniziare una conversazione</li> <li>• Ricevere messaggi</li> <li>• Creare chat di gruppo</li> <li>• Partecipare ad una chat di gruppo</li> <li>• Bloccare/Sbloccare utenti</li> </ul>
Utente non registrato	<ul style="list-style-type: none"> <li>• Visualizzare profilo di un altro utente</li> </ul>	<ul style="list-style-type: none"> <li>• Visualizzare post</li> <li>• Visualizzare dettagli post</li> <li>• Filtrare post per tag</li> </ul>		

### 3.6 Controllo del flusso globale del sistema

Il flusso de software è gestito tramite classi Node.js che elaborano le richieste generate dal Client. Tramite Http, il client scarica le view, e sempre tramite Http, quest'ultime chiedono al web host i dati necessari per mostrare la pagina richiesta dal client. Il web host elabora la richiesta tramite Node.js e ne caso vi sia necessita di accedere a dati persistenti, il web server tramite il connettore mysql2 accede al database server e ottiene i dati che poi restituirà alla view tramite http.

### 3.7 Condizioni boundary

Le condizioni limite che riguardano l'accensione e l'arresto del sistema.

#### 3.7.1 Avvio del sistema

Il sistema dopo l'avvio fornisce a tutti gli utenti collegati ad esso i servizi.

Nome Scenario	SC_3.7.1 Server Startup
Attore	Francesco: Staff member SamelInitialTeam
Flusso di Eventi	<ol style="list-style-type: none"> <li>1. Francesco decide di avviare il sistema quindi preme sul pulsante "avvia"</li> <li>2. Il sistema ricevuta la richiesta avvia i server e esegue dei test per controllare che ogni componente del sistema sia connessa</li> </ol>

ID:	UC_3.7.1
Nome Use Case:	Startup Server
Partecipanti:	<ul style="list-style-type: none"> <li>• Staff member SamelInitialTeam</li> </ul>
Condizioni di entrata:	Il server è pronto per l'avvio
Flusso di eventi:	<p><b>Utente</b></p> <ul style="list-style-type: none"> <li>• Da remoto accede alla macchina sulla quale risiede il server e clicca sulla funzione "Avvia"</li> </ul> <p><b>Sistema</b></p> <ul style="list-style-type: none"> <li>• Il Sistema riceve la richiesta e avvia il server.</li> </ul>
Condizioni di uscita:	Il server è stato avviato
Eccezioni:	Errore nell'avvio di una delle componenti del server



### 3.7.2 Terminazione del sistema

Per la terminazione del sistema è necessario che tutti i sottosistemi siano disattivati. Prima dell'arresto totale del sistema tutte le connessioni vengono terminate e i dati persistenti ancora non memorizzati vengono salvati sul database server, dopodiché anche il database server viene arrestato.

Nome Scenario	SC_3.7.2 Server Shutdown
Attore	Federico: Staff member SameInitialTeam
Flusso di Eventi	<ol style="list-style-type: none"> <li>1. Federico decide di arrestare il sistema, quindi accede alla macchina in remoto e clicca sulla funzionalità "arresta"</li> <li>2. Il server effettua controllo se ci sono eventuali sessioni ancora attive e le termina salvando gli ultimi dati persistenti, disattiva tutti i servizi e comunica con successo l'avvenuta dell'operazione</li> </ol>

ID:	UC_3.7.2
Nome Use Case:	Shutdown Server
Partecipanti:	<ul style="list-style-type: none"> <li>• Staff member SameInitialTeam</li> </ul>
Condizioni di entrata:	Il server è avviato
Flusso di eventi:	<p><b>Utente Sistema</b></p> <ul style="list-style-type: none"> <li>• Da remoto accede alla macchina sulla quale risiede il server e clicca sulla funzione "Arresta"</li> <li>• Il Sistema riceve la richiesta termina eventuali connessioni ancora aperte, salva i dati persistenti e arresta il sistema.</li> </ul>
Condizioni di uscita:	Il server è stato arrestato
Eccezioni:	

### 3.7.3 Fallimento del sistema

Nel caso di errori hardware si cercherà di far sostituire la componente mal funzionante; nell'eventualità in cui si noterà un incremento dell'utenza si provvederà all'acquisto di macchine AWS più potenti.

Nel caso di errori software, il sistema, a seguito della generazione dell'eccezione, non si dovrà arrestare ma al limite dovrà mostrare un messaggio di errore, continuando le sue normali funzioni.

La persistenza dei dati viene gestita dallo stesso servizio di AWS RDS; ci potrebbero essere problemi riguardanti il reperimento dei dati in quanto le macchine sono situate nella East Coast statunitense, quindi il reperimento potrà risultare lento; non dovrebbero esserci problemi riguardo perdite di dati o crash improvvisi della macchina che ospita il DB. Per minimizzare questo rischio, si eseguiranno periodicamente dei backup del database del sistema, mentre la gestione dell'hardware non dipende dal team ma è a cura di Amazon.

## 4. Servizi dei sottosistemi

### 4.1 Gestione profilo utente

Sottosistema	Gestione Autenticazione
Descrizione	Questo sottosistema permette di effettuare tutte le operazioni relative agli account
Servizi offerti	
Servizi	Descrizione
Login	Questo servizio consente l'accesso ad alcuni servizi disponibili solo agli utenti registrati
Logout	Consente di uscire dal sistema

<b>Recupero Password</b>	Questo servizio consente il recupero della propria password per l'accesso al sistema
<b>Cancellazione account</b>	Questo servizio consente la cancellazione del proprio profilo dal sistema
<b>Registrazione nuovo utente</b>	Questo servizio consente ad un utente non registrato di creare un nuovo account per accedere al sistema
<b>Modifica dati di accesso</b>	Questo servizio consente agli utenti che hanno effettuato il login di modificare i propri dati di accesso

<b>Sottosistema</b>	<b>Gestione Biografia</b>
<b>Descrizione</b>	Questo sottosistema permette di effettuare operazioni sul contenuto dei profili personali
<b>Servizi offerti</b>	
<b>Servizi</b>	<b>Descrizione</b>
<b>Inserimento dati personali</b>	Questo servizio permette ad un utente registrato di inserire i dati personali nella propria area utente
<b>Modifica dati personali</b>	Questo servizio permette ad ogni utente registrato di modificare i propri dati personali dall'area utente
<b>Visualizzazione dati personali</b>	Questo servizio consente ad un utente registrato di visualizzare i propri dati personali attraverso un'apposita area personale
<b>Visualizzazione timeline personale</b>	Questo servizio permette allo studente di visualizzare la propria timeline e ai coordinatori di visualizzare la timeline dei propri studenti

## 4.2 Gestione Forum

Sottosistema	Gestione post
Descrizione	Questo sottosistema permette di effettuare tutte le operazioni relative ai post del forum
Servizi offerti	
Servizi	Descrizione
Visualizzazione Post	Questo servizio, disponibile anche agli utenti non registrati, consente di visualizzare tutti i post presenti nel forum in ordine cronologico
Pubblicazione Post	Questo servizio consente di inserire un nuovo post nel forum solo se si ha effettuato l'accesso
Risposta Post	Questo servizio consente di rispondere ad un post solo se si ha effettuato l'accesso
Votazione Post	Questo servizio, disponibile solo per chi ha effettuato l'accesso, consente di votare un post per stabilire un criterio di visibilità/affidabilità
Visualizzazione Post per tag	Questo servizio, disponibile anche agli utenti non registrati, consente di visualizzare i post raggruppati per un determinato tag
Fissa Post	Questo servizio consente, unicamente ai coordinatori, di fissare un post affinché acquisti massima visibilità

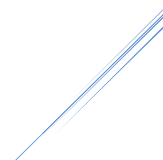
Sottosistema	Gestione annunci
Descrizione	Questo sottosistema permette di effettuare tutte le operazioni relative alla bacheca annunci
Servizi offerti	
Servizi	Descrizione
Visualizzazione Annunci	Questo servizio, disponibile solo agli utenti autenticati, consente di visualizzare la bacheca annunci



<b>Pubblicazione Annunci</b>	Questo servizio consente ai coordinatori di pubblicare un annuncio nella bacheca e di allegare ad esso un file.
------------------------------	---

#### 4.3 Gestione Coordinatori

Sottosistema	Gestione Timeline
<b>Descrizione</b>	Questo sottosistema permette a tutti i coordinatori di gestire e organizzare le esperienze Erasmus degli studenti attraverso le timeline
Servizi offerti	
Servizi	Descrizione
<b>Timeline</b>	Questo servizio permette di avere una panoramica globale dell'esperienza Erasmus dello studente, per capire quali documenti ha consegnato e quando li ha consegnati
<b>Visualizza lista studenti</b>	Questo servizio consente al coordinatore di visualizzare la lista di tutti gli studenti da esso coordinati
<b>Aggiungi studenti</b>	Questo servizio consente al coordinatore di aggiungere uno studente alla propria lista degli studenti coordinati
<b>Tabella ECTS</b>	Questo servizio permette di visualizzare la tabella ECTS necessaria per la conversione dei voti
<b>Matching Esame</b>	Questo servizio permette di generare un suggerimento automatico per l'associazione di un esame con la sua controparte esterna
<b>Matching Voto</b>	Questo servizio permette di generare un suggerimento automatico per la conversione di un voto ottenuto in Erasmus
<b>Upload documenti</b>	Questo servizio permette al coordinatore di caricare dei documenti sulla timeline dello studente.



Sottosistema	Gestione Mappa
Descrizione	Questo sottosistema permette di mostrare una mappa degli studenti attualmente in Erasmus
Servizi offerti	
Servizi	Descrizione
Mappa studenti	Questo servizio permette di generare una mappa europea che mostra la quantità di studenti in Erasmus per ogni località

#### 4.4 Gestione Chat

Sottosistema	Utilizzo chat
Descrizione	Questo sottosistema permette di effettuare tutte le operazioni relative ai post del forum
Servizi offerti	
Servizi	Descrizione
Visualizzazione utenti online/offline	Questo servizio consente agli utenti autenticati di visualizzare gli utenti online ed offline con cui poter messaggiare
Invio messaggio	Questo servizio consente agli utenti autenticati di inviare un messaggio ad altri.
Creazione chat di gruppo	Questo servizio consente agli utenti autenticati di creare chat di gruppo invitando più utenti
Allega file	Questo servizio consente agli utenti autenticati inviare tramite la chat un file.
Blocca utente	Questo servizio consente agli utenti autenticati di bloccare un utente affinché egli non possa più vedere il proprio stato (online o offline) e non possano più ricevere messaggi da lui.

## 5. Glossario

- **Utente:** rappresenta l'utilizzatore del sistema
- **Studente:** rappresenta un utente autenticato che può effettuare diverse operazioni nel sistema ed è interessato all'attività di Erasmus
- **Coordinatore (utente certificato):** rappresenta un utente autenticato che può effettuare diverse operazioni nel sistema ed interagisce con lo studente per gestire la sua attività di Erasmus.
- **Timeline:** rappresenta lo strumento mediante il quale un coordinatore si interfaccia con lo studente per gestire il progresso nell'Erasmus e la validazione dei suoi voti.
- **Node.js:** piattaforma Open source event-driven che consente l'esecuzione di codice JavaScript server side.
- **HTTP:** protocollo di trasferimento di ipertesti che consente a due macchine, client e server, di interagire attraverso un meccanismo di richiesta/risposta.
- **Client:** componente che accede a servizi e risorse offerte dal server.
- **Server:** componente che offre servizi e risorse al client.
- **Model:** componente del modello MVC che si occupa dall'interazione tra l'applicazione e il database.
- **View:** componente del modello MVC scritto in linguaggio HTML e visualizzato nella pagina web dall'utente che permette all'utente di interagire col sistema.
- **Controller:** componente del modello MVC che si occupa di elaborare le richieste di un utente e di comunicare con il model.
- **MVC:** modello architetturale che si basa sull'utilizzo di 3 componenti fondamentali (model,view e controller) per lo sviluppo di web application.
- **HTML:** linguaggio di markup utilizzato per la formattazione e l'impaginazione di documenti ipertestuali.
- **Web host:** servizio di rete che permette di allocare su un server web le pagine di un'applicazione web.
- **Mysql2:** connettore utilizzato da Node.js per l'interazione col database.
- **AWS:** servizio di cloud computing offerto da amazon.
- **AWS RDS:** servizio di AWS che semplifica l'interazione con i database relazionali nel cloud.