

Università degli studi di Salerno

Dipartimento di Informatica

Corso di Laurea in Informatica

Ingegneria del Software

System Design Document

“Ad Maiora”

Versione 1.0

Studenti:

Docente:

Andrea De Lucia

Nome

Collina Ciro

Foudal Soffian

Meriano Davide

Matricola

0512106142

0512105926

0512105958

Anno Accademico: 2020/2021

Indice

1	Introduzione	3
1.1	Scopo del Sistema	3
1.2	Design Goals	3
1.3	Definizioni, acronimi, abbreviazioni	8
1.4	Riferimenti	8
1.5	Panoramica	8
2	Sistema Corrente	8
3	Sistema Proposto	9
3.1	Panoramica	9
3.2	Decomposizione in Sottosistemi	10
3.3	Hardware/Software Mapping	12
3.4	Gestione dei Dati Persistenti	12
3.5	Controllo d'Accesso e Sicurezza	14
3.6	Controllo del Software Globale	15
3.7	Boundary Conditions	15
4	Servizi ai Sottosistemi	20
5	Glossario	22

Data	Versione	Descrizione	Autore
06/01	0.1	Stesura indice, layout e organizzazione del documento	Davide Meriano
09/01	0.4	Definizione ed approfondimento paragrafi	Davide Meriano Soffian Foudal Ciro Collina
10/01	0.5	Correzioni e piccole modifiche al layout	Ciro Collina
11/01	0.7	Completamento del documento	Soffian Foudal
12/01	0.9	Modifiche ai paragrafi	Davide Meriano
19/01	1.0	Piccole modifiche al layout, aggiunta priorità dei Design Goals	Ciro Collina

1 Introduzione

1.1 Scopo del Sistema

La comunicazione è indubbiamente un aspetto importante, per esperienza comune, della crescita personale e, in ambito universitario, accademica. Lo scopo del sistema è di fornire un mezzo di comunicazione online che permetta una facile interazione tra studenti e professori, mantenendo un livello adeguato di formalità. Dal perseguire lo scopo di cui sopra conseguono una serie di vantaggi, inclusi tra gli obiettivi del progetto, che sono diversi, e si focalizzano principalmente sulla comunicazione. Il progetto intende ospitare discussioni virtuali che possano offrire diversi spunti per una crescita accademica, ma anche personale; essendo un sistema web-based, risulta facilmente raggiungibile dagli utenti in qualsiasi periodo dell'anno, offrendo un adeguato supporto didattico anche in periodi dell'anno che non coincidano con il calendario degli insegnamenti. Grazie ad una funzionalità di Q&A, inoltre, s'intende snellire le comunicazioni via mail tra studenti e professori, che costituiscono un canale esclusivo e riservato, sicuramente non adatto a domande e comunicati di carattere generale. Lo scopo del sistema è, inoltre, di migliorare i servizi offerti dall'università, in modo efficiente e non troppo complesso, nel tentativo di aumentarne ulteriormente la qualità.

1.2 Design Goals

Readability :	La leggibilità è un aspetto importante, attorno al quale ruoterà il design di un sistema minimale, ma ben organizzato. Si ritiene che la leggibilità sia un design goal che possa influire anche su perseguimento di altri obiettivi.
Understandability :	Lo sviluppo di un prodotto software è una realtà che notoriamente si interfaccia con diverse figure, che avranno modo di porlo "sotto la lente", ad una scrupolosa analisi. Per favorire tale processo, si pone al primo posto l'understandability, su molti piani differenti. Il goal sarebbe quello di proporre un sistema che possa essere compreso, descritto ed analizzato facilmente.
Ease of Learning :	Il sistema dev'essere intuitivo e le sue funzionalità devono essere facili da imparare, facendo in modo che i nuovi utenti non si sentano "disorientati" durante i primi accessi ed acquisiscano dimestichezza velocemente. S'intende sviluppare il sistema in modo tale che la learning curve dei nuovi utenti sia più bassa possibile.
Ease of Use :	La semplicità di utilizzo è un altro aspetto fondamentale, e si lega facilmente ad altri design

	goals, come il facile apprendimento dell'utilizzo e l'understandability. La facilità con il quale l'utente possa utilizzare il sistema è frutto di un rapporto causa-effetto, dovuto ad altri obiettivi di design, che intendono supportare l'utente in qualità di creatore di contenuti e sui quali verte la piattaforma stessa.
Ease of Remembering :	Nel tentativo di offrire e migliorare il supporto didattico, considerando anche la durata media della "vita accademica" di uno studente, risulta necessaria la consapevolezza che l'utilizzo della piattaforma, soprattutto dal punto di vista studentesco, possa essere discontinuo se non addirittura sporadico; pertanto la facilità con il quale si ricordi il sistema ed il tuo utilizzo è da considerarsi un aspetto di rilievo nonché una diretta conseguenza degli altri obiettivi di design: un sistema intuitivo e facilmente utilizzabile incrementa le probabilità che l'utenza non se ne dimentichi.
User-friendliness :	Il sistema dev'essere user-friendly e presentare interfacce semplici, che mettano a proprio agio l'utente, favorendo l'utilizzo della piattaforma. Questa caratteristica risulta essere molto importante per il look and feel dell'applicazione, che intende offrire funzionalità all'utente preferendo semplicità e naturalezza.
Robusteness :	La robustezza è indubbiamente un concetto chiave: il numero ridotto di funzionalità è controbilanciato dalla loro solidità. Il sistema si propone come maneggevole e concreto, capace di gestire il flusso di esecuzione e gli errori, fornire un'adeguata tracciabilità delle operazioni svolte e dettagliare malfunzionamenti o comportamenti eccezionali. La robustezza getta le fondamenta per una manutenzione agevole ed una gestione reattiva degli errori.
Fault Tolerance :	Tra gli obiettivi del sistema vi è la tolleranza al guasto: s'intende garantire una continuità del servizio evitando interruzioni, intervenendo repentinamente in caso di guasto e cercando di porvi rimedio, laddove possibile. La tolleranza al guasto intende prevenire interventi sul sistema troppo frequenti od invasivi, che precluderebbero l'accesso al sistema ed al suo utilizzo.
Low Cost :	Una caratteristica di spicco del sistema è sicuramente un costo non eccessivo, in termini di risorse, tempo e costi. Si ritiene che questo aspetto

	<p>possa garantire la longevità della piattaforma, e che spiani la strada a sviluppi futuri senza troppe difficoltà, favorendo la crescita del software proposto e la nascita di progetti futuri volti a migliorarlo.</p>
Maintinability :	<p>Anche grazie ai costi contenuti, la manutenibilità del sistema non è da considerarsi un aspetto secondario. Essendo un servizio pensato per ampliare il supporto didattico offerto dall'università, è opportuno che siano rispettati degli standard di manutenibilità, che permettano di limitare il numero di interventi, laddove possibile, e far sì che siano veloci e mirati, onde evitare disservizi.</p>
Rapid Development :	<p>Per questo progetto, considerando le tempistiche ristrette e la necessità di sviluppare il sistema in tempi brevi, la velocità di sviluppo deve ritenersi (in via eccezionale) un design goal prediletto, attraverso il quale sia possibile tamponare il più possibile i ritardi riscontrati nelle fasi precedenti della progettazione ed assicurare una consegna del progetto in tempo utile.</p>

Di seguito è riportata un'analisi dei trade-off più comuni, con relativa (ndr. breve) argomentazione, allo scopo di motivare le scelte di design, di cui questo documento, e fornire una breve panoramica sul pensiero che ne abbia gettato le fondamenta.

- **Functionality vs. Usability**

La natura del sistema suggerirebbe una predilezione per l'usabilità di per sé, ma non è l'unico motivo per cui in questo progetto l'usabilità prevalga sulle funzionalità. Questa scelta deriva molto anche dalle aspettative nei confronti del prodotto finale, che in maniera discretamente visionaria, sarebbe incentrato sulla semplicità, su uno sfruttamento naturale ed intuitivo della funzionalità offerte, ponendo al centro l'utente e cercando di assisterlo adeguatamente, nei limiti del possibile.

- **Cost vs. Robustness**

Il trade-off Costi-Robustezza si risolve con una chiara predilezione per la Robustezza, poiché una caratteristica di punta del sistema è la continuità del flusso di esecuzione, e la capacità di gestire comportamenti erronei del sistema indotti dall'input dell'utente. Si è scelto di tralasciare i costi poiché si stima un costo molto basso, da previsioni approssimative, almeno nelle prime versioni del sistema, pertanto è un aspetto che non ha influenzato particolarmente le scelte di progetto.

- **Efficiency vs. Portability**

Il trade-off efficienza-portabilità, per questo progetto, vedrebbe favorita l'efficienza, poiché si preferirebbe un sistema funzionante, che permetta l'offerta di funzionalità ben strutturate, capaci di scaturire un workflow fluido e non eccessivamente limitato. Essendo una piattaforma web-based, è considerevole l'apporto che la facile accessibilità al sistema garantita agli utenti fornisca alla portabilità. La scelta tra efficienza e portabilità risulta dunque particolarmente ostica: la portabilità è una conseguenza della natura del sistema, tuttavia, in fase di sviluppo sarebbe preferibile optare per l'efficienza. In virtù delle valutazioni di cui sopra, e considerando la portabilità come una caratteristica innata del software in oggetto, si può affermare che le scelte di progetto volgono più per l'efficienza.

- **Rapid Development vs. Functionality**

La scelta tra velocità di sviluppo e funzionalità vede largamente favorita la velocità di sviluppo: si è preferito offrire un set minimale di funzionalità per diversi motivi, tra cui la velocità di sviluppo. In questa fase del progetto, analizzando anche i costi in funzione del tempo, la velocità con il quale si possa sviluppare funzionalità, anche in fasi successive (ndr. eventuale ampliamento delle funzionalità offerte), la velocità con il quale si possa intervenire sul codice è un aspetto prioritario.

- **Cost vs. Reusability**

Il trade-off costi-riusabilità, considerando che i costi siano di per sé relativamente bassi, vede favorita la riusabilità. S'intende sviluppare un software che possa dare inizio ad un servizio offerto dall'università, che come tale abbia delle fondamenta solide, sulle quali basare eventuali sviluppi futuri. Il riutilizzo faciliterebbe notevolmente lo sviluppo rapido, e sarebbe inoltre una caratteristica ideale per spianare la strada alla crescita del software, preservando adeguatamente tempi e costi.

- **Backward Compatibility vs. Readability**

Essendo un progetto di greenfield engineering, si può affermare con certezza che la retrocompatibilità del software non fosse un aspetto prioritario. Una naturale tendenza della progettazione, in questo caso, è stata quella di volgere verso la leggibilità, anche basandosi sulle aspettative nei confronti del prodotto finito. Si ritiene che la leggibilità sia un aspetto importante, che possa favorire futuri miglioramenti, una crescita agevole delle funzionalità, ed uno sviluppo meno difficoltoso possibile.

Design Goals' Priority

Di seguito è riportata, in modo schematico, un'analisi di priorità dei Design Goal, cercando di descrivere ogni aspetto del design in funzione di quanto verrà preso in considerazione durante i trade off nelle fasi successive della progettazione e sviluppo del progetto. La tabella seguente consta dell'associazione tra un design goal ed un indice di priorità che ha un range da uno a cinque, dove cinque indica che si prenderà fortemente in considerazione l'obiettivo associato. Non avendo citato tutti i design goal, bensì quelli di maggiore importanza, valutati più approfonditamente nella stesura del documento, i valori di priorità tenderanno ad essere mediamente più alti, rispetto ad una tabella completa.

Design Goal	Priority
Readability :	Quattro
Understandability :	Quattro
Ease of Learning :	Quattro
Ease of Use :	Cinque
Ease of Remembering :	Cinque
User-Friendliness :	Cinque
Robustness :	Tre
Fault Tolerance :	Tre
Low Cost :	Due
Maintainability :	Tre
Rapid Development :	Quattro
Functionality :	Tre
Usability :	Cinque
Cost :	Due
Robustness :	Tre
Efficiency :	Tre
Reusability :	Tre
Backward Compatibility :	Uno

1.3 Definizioni, acronimi, abbreviazioni

"Mod."	: Moderatore, figura coinvolta nella gestione dei contenuti del forum.
"Q&A"	: Question and Answer – Domanda e Risposta.
"N.A."	: Not Applicable – Danote l'assenza di una risposta rilevante.
"ndr."	: Nota del Redattore, Nota della Redazione.
"ER"	: Entity-Relationship – Entità-Associazione.
"DBA"	: DataBase Administrator – Amministratore della Base di Dati.
"DBMS"	: DataBase Management System – Sistema di Gestione della Base di Dati.
"SA"	: System Administrator – Amministratore di Sistema.
"GUI"	: Graphical User Interface – Interfaccia Grafica.
"UCBC"	: Boundary Condition's Use Case – Caso d'Uso della condizione limite.
"BS"	: Boundary Scenario – Scenario limite, Scenario di una condizione limite.

1.4 Riferimenti

- Slide del Docente (reperibili sulla piattaforma).
- Requirements Analysis Document v2.0, progetto "Ad Maiora".
- B.Bruegge, A.H. Dutoit, Object Oriented Software Engineering - Using UML, Patterns and Java, Prentice Hall, 3rd edition, 2009.

1.5 Panoramica

Il progetto nasce per estendere i servizi online dell'Università degli Studi di Salerno, pertanto intende collocarsi in un contesto universitario, ed offrire funzionalità ad un gruppo tanto vasto quanto vario di individui. In questa fase di design, l'intento sarebbe quello di stabilire dei criteri validi per una prima versione del sistema che non esclude sviluppi futuri ed eventuali redesign. Il design si concentra principalmente sulla definizione di un sistema poco costoso, capace di offrire un set minimale di funzionalità e garantirne il funzionamento. Proseguendo nel documento è possibile individuare tutte le modalità ed i ragionamenti utilizzati per il design del sistema, stabilendo chiaramente le scelte intraprese a livello d'architettura, con l'obiettivo di garantire un riferimento del sistema in tutti i suoi aspetti.

2 Sistema Corrente

"Ad Maiora" è frutto di un processo di greenfield engineering.

Un sistema simile a quello proposto è "Reddit": un forum, un sito web di discussione, aggregato di social news e sito di web content rating americano, altresì noto come "La prima pagina di Internet" (ndr. accezione giornalistica). La registrazione su Reddit è gratuita e non richiede l'utilizzo di mail. Ogni utente può commentare e votare il contenuto generato da altri utenti, gli utenti registrati

hanno accesso alla creazione di un proprio subreddit su un argomento scelto. I subreddit sono monitorati da moderatori, utenti di reddit che ne guadagnano il titolo. Generalmente, all'interno del sito, un moderatore è un volontario che gestisce le proprie community, stabilisce e fa rispettare delle regole specifiche (che possono differire a seconda della community), rimuovere post e commenti che le violano, e cercano di mantenere la discussione in linea con gli argomenti trattati. Una figura contrapposta a quella dei moderatori è l'admin, che si ritiene essere a tutti gli effetti un dipendente di Reddit, pagato per il suo lavoro. Reddit inoltre ha introdotto un sistema di membership a pagamento e un sistema di valuta virtuale, capaci di differenziare ulteriormente gli utenti ed il loro accesso a funzionalità diverse. Reddit era inizialmente scritto in Common Lisp, prima di sviluppare le proprie librerie open source per lo sviluppo web ed essere riscritto in Python. Il sistema ha subito una migrazione che ha portato alla disconnessione dei server proprietari in favore degli Amazon Web Services. La principale risorsa per la gestione dei dati persistenti è PostgreSQL, tuttavia sta lentamente volgendo all'uso di Apache Cassandra.

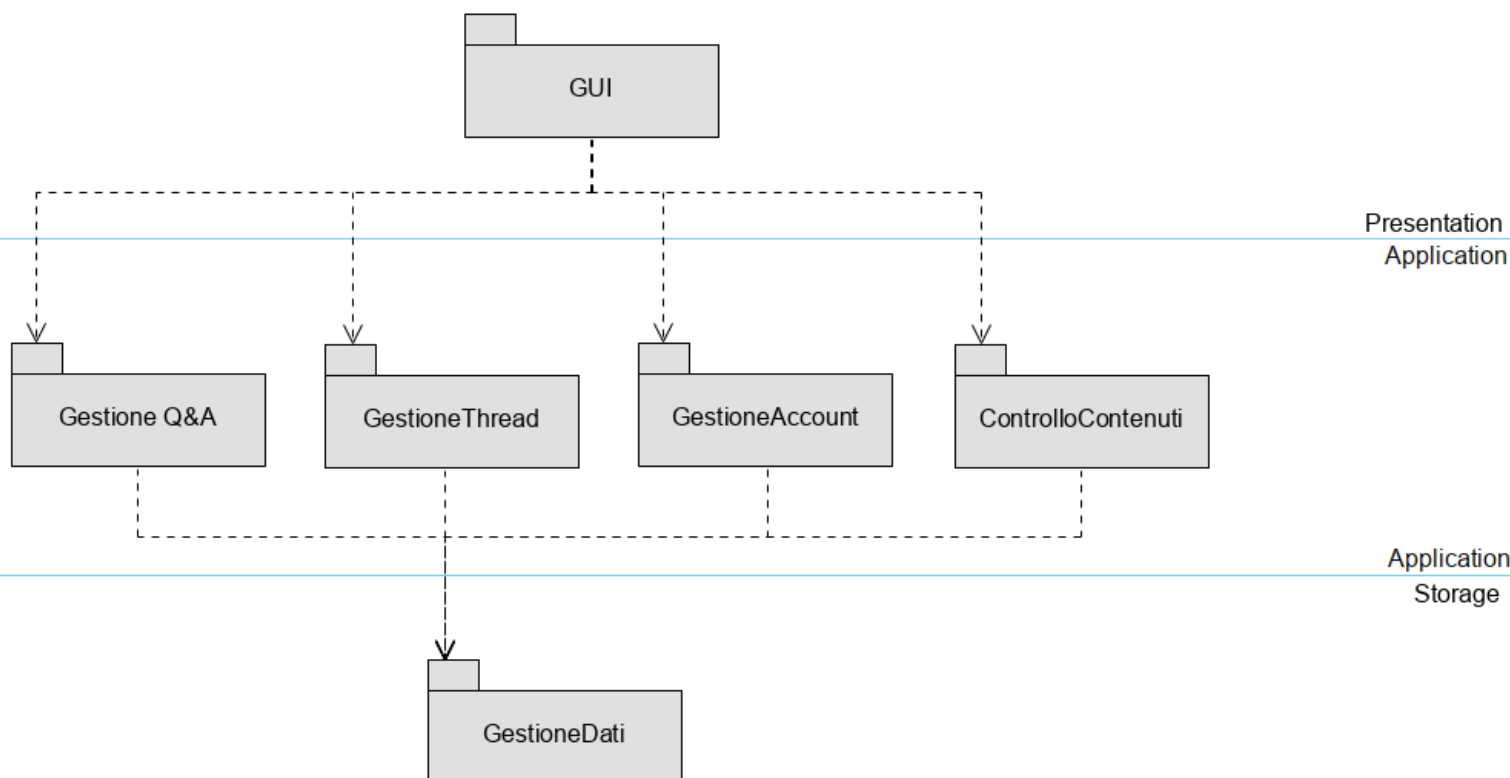
Il design del sistema è cambiato innumerevoli volte col passare del tempo, ma è possibile dedurre, anche basandosi sulle funzionalità implementate successivamente, che tra gli obiettivi di Reddit ci sia l'usabilità della piattaforma, che deve tuttavia confrontarsi con il numero elevato di funzionalità offerte ed in via di sviluppo. Risulta inoltre evidente che con la crescita esponenziale del sistema, e la sua popolarità, un aspetto prioritario del sito sia la performance aspetto che ha indotto gli sviluppatori ad abbandonare l'utilizzo di diversi software di terze parti, preferendo componenti customizzati: scelta che comporta un incremento notevole dei costi. Negli anni, Reddit ha subito diversi redesign, concentrati principalmente sull'estensione della piattaforma, conciliando all'intraprendenza imprenditoriale un rispetto dei propri obiettivi di design.

3 Sistema Proposto

3.1 Panoramica

Il sistema è un forum testuale basato interamente sul contenuto user-created, che può assumere diverse forme ed al quale si può interagire attraverso diverse funzionalità. Il software sarà reso accessibile attraverso Internet e sarà raggiungibile da tutti i dispositivi dotati di una connessione ad internet ed un web browser. La piattaforma è suddivisa in sottosistemi organizzati in tre livelli, che permettono di differenziare e ripartire adeguatamente le responsabilità necessarie a garantire l'offerta delle funzionalità, in un ambiente ordinato e sicuro. Di seguito, sarà possibile approfondire tutti gli aspetti legati al sistema, dal punto di vista hardware e software, fornendo spunti su tutto ciò che interagisce col sistema.

3.2 Decomposizione in Sottosistemi



- **Design Architecture**

L'architettura scelta per il design è di tipo Model View Controller, e inquadra tre tipologie di sottosistemi. Questo genere di architettura è particolarmente adatto per applicazioni Web interattive, favorisce il dinamismo, e può considerarsi una tra le architetture più appropriate per sistemi medio-grandi (ndr. le dimensioni di questo progetto sono nettamente inferiori). In virtù di questa scelta, la separazione tra logica di business e logica di presentazione è molto netta.

- **Layers**

I sottosistemi sono suddivisi in tre layer differenti: Storage, Presentation e Application.

Lo Storage Layer rappresenta il livello dei dati, incluse le operazioni per accesso e modifica. Lo Storage deve notificare i componenti associati quando viene modificato e deve supportare la possibilità per il Presentation di interrogare lo stato dei dati, e all'Application Layer la possibilità di accedere alle funzionalità incapsulate al suo interno.

Il Presentation Layer si occupa del rendering dei contenuti: accede ai dati tramite lo Storage e specifica come i dati debbano essere presentati. Esso si occupa principalmente di aggiornare la presentazione dei dati quando il loro stato cambia ed inviare all'Application Layer l'input sottomesso

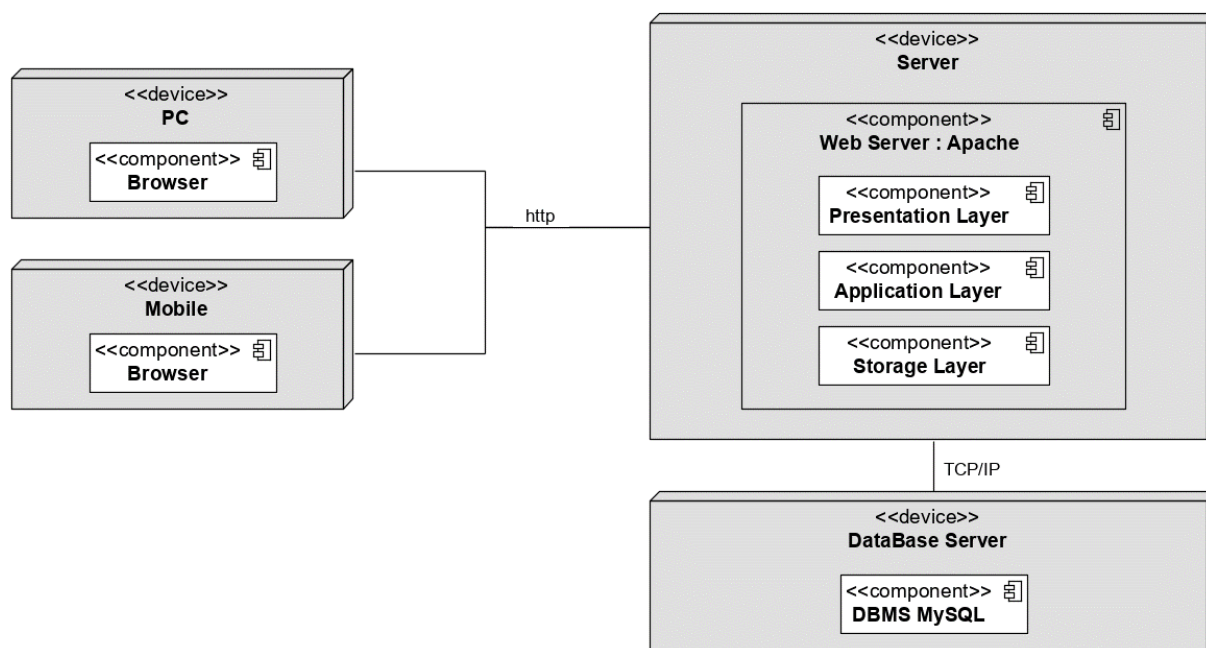
dall'utente. Si occupa di fare il dispatching delle richieste provenienti dall'utente e selezionare la presentazione più adatta: lo scopo principale del Presentation Layer è quello di interpretare l'input dell'utente e mapparla su azioni che devono essere eseguite dall'Application. L'Application Layer definisce il comportamento dell'applicazione, si occupa dell'esercizio delle funzionalità del sistema. Questo layer prende in consegna l'input utente fornito dal Presentation Layer e lo sfrutta per applicare la logica di business, ha accesso allo Storage Layer ed è in grado di ricondurre una richiesta da parte dell'utente ai dati associati attraverso operazioni ben definite.

Subsystems

La divisione, anche in funzione dell'architettura scelta, ha prodotto i seguenti sottosistemi, organizzati in layer come riportato nel grafico di cui sopra.

GestioneDati	:	Il sottosistema si occupa della rappresentazione dei dati, e fornisce accesso alle operazioni CRUD attraverso le quali gestirli.
GUI	:	Il sottosistema si occupa della presentazione dei dati all'utente, nonché di una prima gestione dell'input.
GestioneAccount	:	Il sottosistema si occupa della gestione della logica per quanto riguarda le credenziali, l'autenticazione, e la gestione dell'accesso degli utenti.
GestioneThread	:	Il sottosistema si occupa della gestione della logica di business per quanto riguarda la gestione delle discussioni virtuali. Che nascono e si sviluppano sulla piattaforma.
GestioneQ&A	:	Il sottosistema si occupa della gestione della logica di business per quanto riguarda le funzionalità di domanda e risposta offerte dal sistema.
ControlloContenuti	:	Il sottosistema si occupa della gestione della logica di business per quanto riguarda le funzionalità amministrative dedicate ai moderatori.

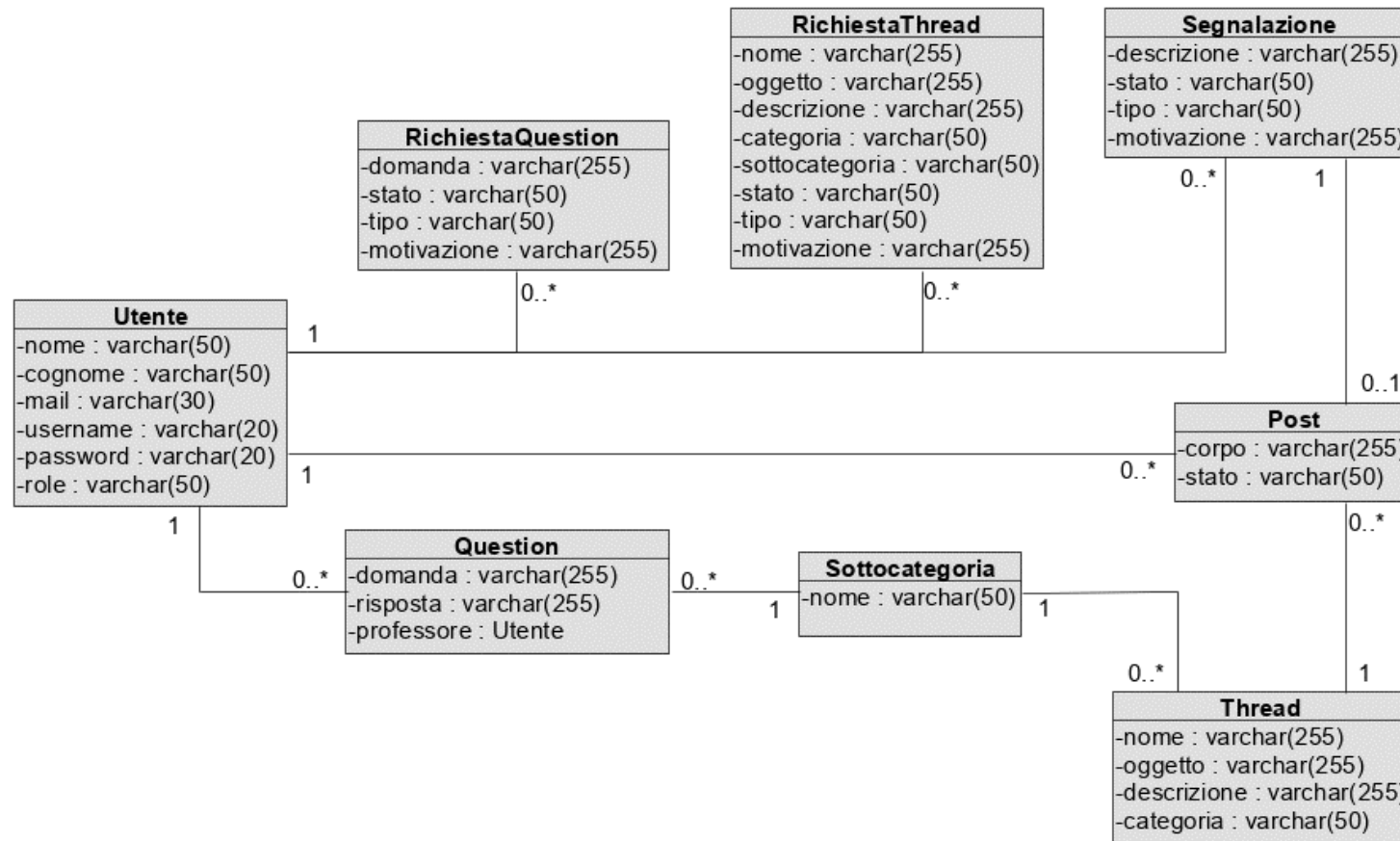
3.3 Hardware/Software Mapping



Il sistema che s'intende sviluppare utilizzerà una struttura hardware costituita da un Server, in grado di rispondere alle richieste del Client utilizzando il protocollo http. Sul server verrà effettuato il deployment dell'applicazione, che la renderà disponibile ed accessibile a tutti. Il Server si interfacerà, mediante protocollo TCP/IP con il DataBase Server, a cui è destinata la gestione della persistenza dei dati, gestita grazie ad un DataBase Management System MySQL. Il sistema sarà accessibile da diversi dispositivi, in particolare, qualsiasi dispositivo dotato di una connessione ad Internet e di un web browser, attraverso i quali raggiungere il Server.

3.4 Gestione dei Dati Persistenti

Per la gestione dei dati persistenti si intende optare per una Base di Dati relazionale, associata al sistema, con DBMS MySQL. Di seguito è riportato il diagramma ER che si ritiene riflettere al meglio la struttura attraverso la quale memorizzare e gestire i dati persistenti.



3.5 Controllo d'Accesso e Sicurezza

Attori	Studente	Professore	Moderatore delle Richieste	Moderatore delle Segnalazioni
Accesso				
LoginForm :	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit
PaginaPersonale :	✓ Modifica dati	✓ Modifica Dati ✓ Lista Domande	✓ Modifica Dati ✓ Lista Richieste	✓ Modifica Dati ✓ Lista Segnalazioni
CreaThread_Form :	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit		
VisualizzaRichiesta_Form :			✓ Approva Richiesta ✓ Rifiuta Richiesta ✓ Compila Dati ✓ Submit	✓ Approva Segnalazione ✓ Rifiuta Segnalazione ✓ Compila Dati ✓ Submit
PubblicaPost_Form :	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit		
SegnalaPost_Form :	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit		
CreaQuestion_Form :	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit		
VisualizzaQuestion_Form :		✓ Compila Dati ✓ Submit		
Registrazione_Form :	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit
VisualizzaThread_Pagina :	✓ "Pubblica" ✓ "Segnala" ✓ Compila Dati ✓ Submit	✓ "Pubblica" ✓ "Segnala" ✓ Compila Dati ✓ Submit		
RisultatiRicerca_Pagina :	✓ Click	✓ Click		
VisualizzaQ&A_Pagina :	✓ Click ✓ "Domanda"	✓ Click		
ModificaProfilo_Form :	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit	✓ Compila Dati ✓ Submit

"Ad Maiora" è un sistema multi-utente, che prevede diversi ruoli facilmente deducibili in fase di registrazione. Le funzionalità offerte ai ruoli sono molto simili, seppur differenziate. Per il ruolo di "studente" è previsto un accesso libero alle funzionalità di pubblicazione, ad eccezione delle risposte nelle sezioni Q&A, riservate ai professori nonché unico punto di disgiunzione tra gli accessi dei due ruoli sopracitati. La figura di moderatore è incentrata molto di più sull'aspetto amministrativo e relativa gestione. La base della tassonomia dei moderatori è una divisione per contenuti: si avvertiva la necessità che ogni tipologia avesse una figura amministrativa di riferimento. Analizzando la documentazione, si può notare tuttavia che, il ruolo di Moderatore delle Richieste sia quello destinato allo svolgimento di più compiti diversi, rispetto

all'altra figura: questo perché, da alcune previsioni della crescita del software, ci si aspetta che la domanda di creazione di thread, seppur non immediatamente, tenda a calare in funzione del tempo; il numero di domande nelle sezioni Q&A invece si prevede sia un numero limitato, ma costante nel tempo. Il trend si prospetta diametralmente opposto per quanto riguarda il numero di segnalazioni nel tempo: con la crescita del sistema, il numero di post pubblicati potrebbe aumentare esponenzialmente, incrementando considerevolmente il numero di segnalazioni e di conseguenza il carico di lavoro dei relativi Moderatori (assumendo che gli utenti abbiano cura dell'integrità delle discussioni e rispetto delle buone norme di comunicazione). Si ritiene che questa ripartizione dei compiti, differenziando nettamente le figure amministrative per tipologia di contenuti, possa facilitare la crescita del sistema e lo sviluppo di nuove funzionalità.

3.6 Controllo del Software Globale

Il flusso globale del sistema, e la serie di funzionalità che lo costituiscono, dipende strettamente da interazioni con l'utente, che effettua richieste tramite un web browser. Il flusso globale del sistema è event-driven, e sfrutta eventi generati da una GUI attraverso input come click e selezione del menu.

3.7 Boundary Conditions

- **Configuration**

Gli oggetti persistenti all'interno del sistema rappresentano principalmente contenuti user-created, pertanto il loro ciclo di vita è dettato dal comportamento delle varie figure che utilizzano il sistema. Si ritiene, tuttavia, che presentare all'utente una piattaforma priva di contenuti sarebbe sicuramente conseguenza di una configurazione erranea e sarebbe opportuno presentare all'utente discussioni predefinite, alle quali poter direttamente aggiungere post. In particolare, l'obiettivo della configurazione del sistema sarebbe quello di creare dei thread generici, uno per ogni corso del dipartimento, in particolare, che rappresentino una generalizzazione di tutte le possibili discussioni future che possano nascere spontaneamente dagli utenti. Conoscendo gli oggetti di tipo Thread, e sapendo che ognuno sia dotato di una categoria, una sottocategoria, un nome, un oggetto ed una descrizione, il formato più adatto dei thread predefiniti potrebbe essere il seguente:

Thread	
Categoria	: Nome del Dipartimento
Sottocategoria	: Nome del Corso
Nome	: Nome del Corso
Oggetto	: "General"
Descrizione	: "Discussione generica, riferita al corso di laurea..."

Una diretta conseguenza della configurazione dei thread sarebbe la creazione di tutte le possibili sottocategorie, aspetto importante ai fini dell'ordine dei contenuti per il quale viene riservato all'utente il solo diritto di scelta, in fase di richiesta di creazione di un thread. Un'altra importante conseguenza di questa configurazione è la creazione automatica di una sezione Q&A associata al thread, per la quale non si ritengono necessarie ulteriori configurazioni. Ad eccezione di questo intervento preliminare sui dati memorizzati, non sono previste ulteriori configurazioni di alcun genere.

La configurazione dei dati persistenti che non sono gestiti nei casi d'uso precedentemente definiti è una responsabilità del DataBase Administrator, sotto la supervisione del System Administrator.

Scenari

BS ₁ :	Andrea è il DataBase Administrator della base di dati del sistema "Ad Maiora", e stando alle direttive del System Administrator, bisogna intervenire sui dati persistenti del sistema allo scopo di fornire dei thread predefiniti. Andrea, utilizzando il DBMS della base di dati associata al sistema, provvedendo all'inserimento, nota che la sottocategoria del thread che deve creare non esiste: imposta il nome a "Ingegneria del Software" e la memorizza. Dopo aver inserito la sottocategoria, utilizzando la tabella definita dal System Design Document del sistema come riferimento, inserisce tutti i dati del thread "General" relativo al corso di Ingegneria del Software. Andrea provvede a crearlo, ed inserisce come categoria: "DInf", come nome: "Ingegneria del Software", come oggetto: "General" e come descrizione: "Discussione generale, relativa al corso di Ingegneria del Software del Dipartimento di Informatica". Dopo aver inserito questi dati, associa il thread alla sottocategoria precedentemente creata. Andrea ripete il processo per ogni corso del Dipartimento di Informatica che non dispone ancora di una discussione "General" all'interno del sistema "Ad Maiora".
-------------------	--

Casi d'Uso

ID:	UCBC_1
Use Case Name:	Configurazione Dati
Participating Actors:	DataBase Administrator
Entry Condition:	<ul style="list-style-type: none"> Non esiste un thread "General" di riferimento per un corso, memorizzato all'interno del sistema
Flow of Events:	
DBA :	Il DBA accede alla Base di Dati del sistema ed inserisce i dati relativi alla sottocategoria del thread che intende memorizzare.

DBMS :	Il DBMS coordina l'inserimento e convalida la creazione di una sottocategoria.
DBA :	Il DBA provvede all'inserimento dei dati relativi al thread "General" che intende creare, e vi associa la sottocategoria precedentemente definita.
DBMS :	Il DBMS coordina l'inserimento e convalida la creazione del thread specificato.
Exit Condition:	Il thread è stato creato, consta delle informazioni specificate dall'attore ed è accessibile tramite le funzionalità del sistema.
Exception:	
La Sottocategoria del thread esiste ed è memorizzata	Se la sottocategoria del thread che si intende creare è già memorizzata, l'attore provvede semplicemente ad associarvi il thread che intende memorizzare.

Initialization Scenari

BS₂:	Gianni è l'amministratore del sistema "Ad Maiora", e deve inizializzare il sistema ed effettuare lo startup del server. Gianni accede al sistema e clicca sull'apposito pulsante "Start". Il sistema inizia una procedura di inizializzazione, dettagliando tutti gli step sulla console. Al termine della procedura, l'ultimo messaggio notifica il successo dell'operazione all'amministratore, che sa con certezza che il sistema sia accessibile agli utenti ed offra tutte le sue funzionalità.
------------------------	--

Casi d'Uso

ID:	UCBC_2
Use Case Name:	Startup Server
Participating Actors:	System Administrator
Entry Condition:	· L'amministratore accede al sistema.
Flow of Events:	
SA :	L'amministratore clicca sul pulsante "Start", dopo aver acceduto al sistema.
Sistema :	"Ad Maiora" attiva i propri servizi rendendosi disponibile agli utenti e notifica all'amministratore il successo dell'operazione.
Exit Condition:	Il sistema è attivo, ed offre tutte le sue funzionalità.

Exception:	
Startup Error	Nel caso in cui si manifestassero errori di startup, il sistema non sarebbe nelle condizioni di funzionamento ideale, e sarebbe a discrezione dell'amministratore scegliere come procedere in tal caso. Potrebbero essere opportuni degli interventi di manutenzione, col fine di riportare il sistema ad uno stato solido con il successo della procedura di startup.

· Termination Scenari

BS₃:	Gianni è l'amministratore del sistema "Ad Maiora", e deve effettuare la terminazione del server. Gianni accede al sistema e clicca sull'apposito pulsante "Arresta". Il sistema inizia una procedura di terminazione, che comprende la disconnessione di eventuali client connessi, dettagliando sulla console l'esito di tutti gli step. Al termine della procedura, l'ultimo messaggio notifica all'amministratore il suo successo, che sa con certezza che il sistema sia offline e non più accessibile.
------------------------	---

Casi d'Uso

ID:	UCBC_3
Use Case Name:	Shutdown Server
Participating Actors:	System Administrator
Entry Condition:	· L'amministratore accede al sistema.
Flow of Events:	
SA :	L'amministratore clicca sul pulsante "Arresta", dopo aver acceduto al sistema.
Sistema :	"Ad Maiora" verifica la presenza di client connessi al sistema, li disconnette ed avvia la procedura d'arresto. Al termine dell'operazione, notifica il suo successo.
Exit Condition:	Il sistema è attivo, ed offre tutte le sue funzionalità.
Exception:	
Shutdown Error	Potrebbero incorrere dei problemi nella procedura di shutdown, che verrebbero opportunamente notificati. In tal caso sarebbe una responsabilità dell'amministratore la scelta di come procedere, valutando l'opzione di una terminazione forzata.

- **Failure**

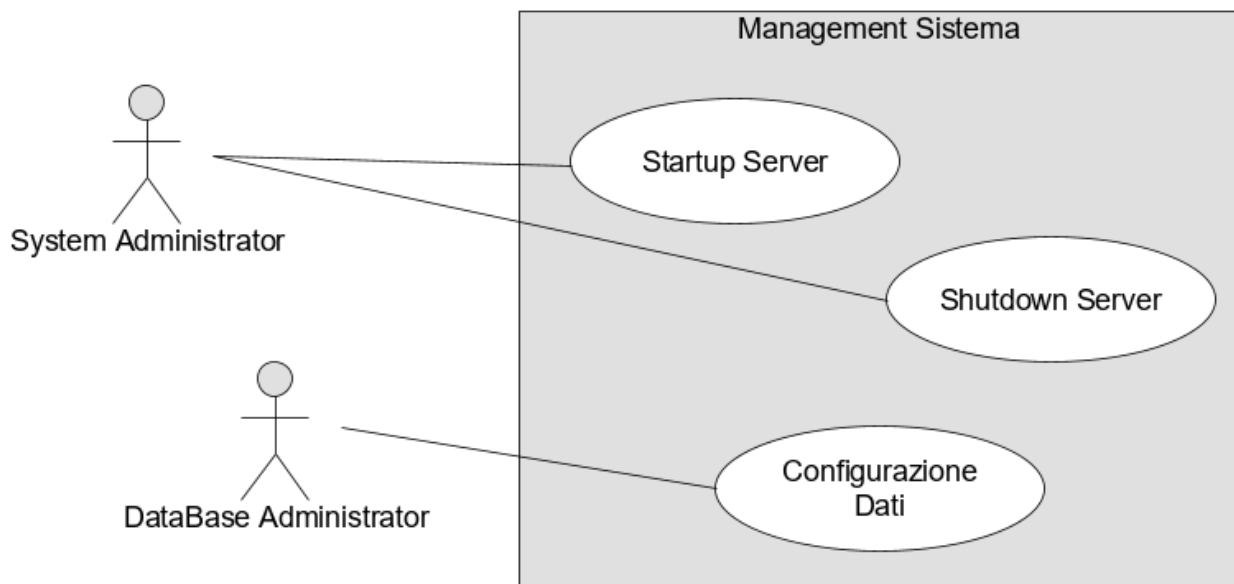
In caso di guasti dovuti al sovraccarico del DataBase con successivo fallimento dello stesso, è previsto un salvataggio di script che permettano di recuperarne la struttura. Non si ipotizza un recupero dei dati memorizzati, poiché non sono previste operazioni di backup periodico (ndr. non si esclude che in fasi successive possano essere integrate con successo).

In caso di brusche ed inaspettate interruzioni dell'alimentazione non sono previsti metodi per il ripristino di uno stato ideale del sistema precedente allo spegnimento eccezionale (ndr. tuttavia si potrebbero integrare con l'utilizzo di gruppi di continuità che permettano, in caso di cessazione di corrente, un salvataggio dello stato del sistema prima dello shutdown in sicurezza).

In caso di malfunzionamenti dovuti al comportamento del sistema non sono previste politiche d'intervento che non coinvolgano un team di manutenzione, ad eccezione di un riavvio del sistema, nell'ipotesi che il comportamento erroneo sia indotto da malfunzionamenti non critici.

Un altro caso di fallimento potrebbe essere dovuto a malfunzionamenti critici dell'hardware ai quali non sarebbe possibile porre rimedio, seppure tali condizioni siano monitorabili con controlli periodici da parte di un team di manutenzione.

- **Boundary Conditions' Use Case Model**



4 Servizi ai Sottosistemi

· UtenteGUI

Servizio: Pubblicazione Post

Descrizione: Permette la pubblicazione di un post.

Servizio: Segnalazione Post

Descrizione: Permette di segnalare un post.

Servizio: Richiesta di creazione di un thread

Descrizione: Permette di richiedere la creazione di una discussione.

Servizio: Richiesta di creazione question

Descrizione: Permette di richiedere che una domanda specifica sia sottoposta ad un professore.

Servizio: Pubblicazione Risposta

Descrizione: Permette di rispondere ad una question.

Servizio: Ricerca Contenuti

Descrizione: Permette la ricerca di contenuti pubblici.

Servizio: Modifica Dati Profilo

Descrizione: Permette di modificare le informazioni relative al proprio account.

Servizio: Login

Descrizione: Permette l'autenticazione al sistema.

Servizio: Registrazione

Descrizione: Permette la creazione di un account.

Servizio: Navigazione

Descrizione: Permette di navigare il sito, cambiare pagine ed interagire con altri servizi.

Servizio: Gestione Richieste

Descrizione: Permette di visualizzare, accettare e rifiutare richieste.

Servizio: Gestione Segnalazioni

Descrizione: Permette di visualizzare, accettare e rifiutare segnalazioni.

- **GestioneAccount**

Servizio: Login

Descrizione: Coordina l'autenticazione al sistema.

Servizio: Modifica Dati Profilo

Descrizione: Coordina la modifica delle informazioni relative al proprio account.

Servizio: Registrazione

Descrizione: Coordina la creazione di un account.

- **GestioneThread**

Servizio: Richiesta di creazione di un thread

Descrizione: Coordina le richieste di creazione di discussioni.

Servizio: Ricerca Contenuti

Descrizione: Coordina la ricerca di thread pubblici.

Servizio: Pubblicazione Post

Descrizione: Coordina la pubblicazione di post.

Servizio: Segnalazione Post

Descrizione: Coordina la segnalazione di post.

- **GestioneQ&A**

Servizio: Richiesta di creazione question

Descrizione: Coordina le richieste di creazione di una question.

Servizio: Pubblicazione Risposta

Descrizione: Coordina la risposta ad una question specifica.

Servizio: Ricerca Contenuti

Descrizione: Coordina la ricerca di sezioni Q&A.

- **ControlloContenuti**

Servizio: Gestione Richieste

Descrizione: Coordina la gestione delle richieste di creazione di thread.

Servizio: Gestione Segnalazioni

Descrizione: Coordina la gestione delle segnalazioni.

GestioneDati

Servizio: Gestione Account

Descrizione: Gestione della memorizzazione degli account.

Servizio: Gestione Thread

Descrizione: Gestione della memorizzazione delle discussioni.

Servizio: Gestione Post

Descrizione: Gestione della memorizzazione dei post.

Servizio: Gestione Question

Descrizione: Gestione della memorizzazione delle domande (Q&A).

Servizio: Gestione Answer

Descrizione: Gestione della memorizzazione delle risposte (Q&A).

Servizio: Gestione Richieste

Descrizione: Gestione della memorizzazione delle richieste.

Servizio: Gestione Segnalazioni

Descrizione: Gestione della memorizzazione delle segnalazioni.

5 Glossario

"Ad Maiora"	:	Il nome del gruppo di progetto, nonché quello del sistema che si intende progettare. Con questo termine, all'interno del documento, ci si riferisce unicamente al sistema. Talvolta le due parole che compongono il termine non vengono separate da spazio: ciò non altera la semantica del termine.
"Thread"	:	Con il termine "thread" ci si riferisce ad una discussione tra utenti all'interno del sistema. Un thread aggrega dei post.
"Post"	:	Con il termine "post" si intende un messaggio, di natura testuale, pubblicato da un utente all'interno di una specifica discussione.
"Q&A"	:	Con il termine "Q&A", talvolta preceduto da "Sezione", s'intende l'area del sistema all'interno della quale è possibile visualizzare le domande e le risposte degli utenti, in merito ad una categoria specifica.
"Question"	:	Una "question" è una domanda, che uno studente può sottoporre al professore desiderato. È un contenuto testuale che si può trovare in una specifica Sezione Q&A del sito.

"Answer"	:	La risposta di un professore ad una domanda a lui destinata. È un contenuto testuale che si può trovare all'interno di una specifica Sezione Q&A.
"Richiesta"	:	Il termine "Richiesta" viene utilizzato per indicare una compilazione dell'utente all'interno del sistema, grazie alla quale è possibile sottoporre un contenuto ad un moderatore, che può eventualmente approvarne la pubblicazione. Il termine viene associato indistintamente a thread e question, che passano entrambi una fase di pre-pubblicazione. Talvolta, nel documento, ci si riferisce ad una "Richiesta" anche con il termine "Richiesta di Creazione".
"Segnalazione"	:	Il termine "Segnalazione" viene utilizzato per indicare la compilazione di un utente che porta un post pubblico all'attenzione di un moderatore, il quale potrà eventualmente decidere di ripubblicarlo o eliminarlo da una discussione. Talvolta si intende lo stesso, impropriamente, col termine "Richiesta di Segnalazione".
"Utente"	:	Con il termine "Utente" si fa riferimento ad un qualsiasi utilizzatore del sistema, il cui ruolo sia "Studente" o "Professore" (ndr. ma non entrambi).
"Moderatore delle Richieste"	:	Con il termine "Moderatore delle Richieste" s'intende quella figura del lato amministrativo incaricata di supervisionare i contenuti per i quali esista una fase di pre-pubblicazione (thread e question). Il termine va a sostituire completamente quello utilizzato in una fase precedente della progettazione: "Moderatore del Flusso di Creazione".
"Mod delle Segnalazioni"	:	Con il termine "Moderatore delle Segnalazioni" s'intende quella figura del lato amministrativo incaricata di supervisionare le segnalazioni. Il termine va a sostituire completamente quello utilizzato in una fase precedente della progettazione: "Moderatore del flusso di Contenuti".