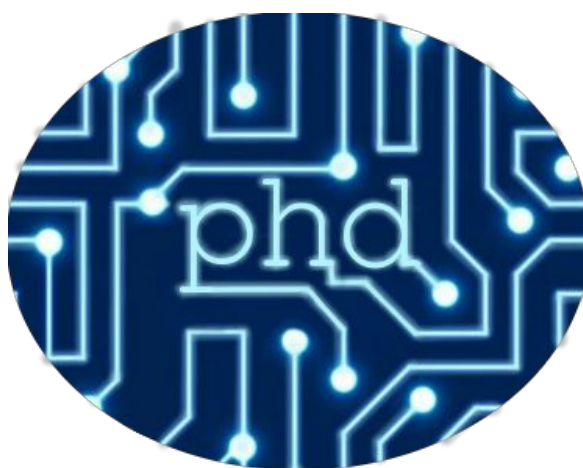




UNIVERSITA' DEGLI STUDI DI SALERNO
FACOLTA' DI SCIENZE MM.FF.NN.
CORSO DI LAUREA TRIENNALE IN INFORMATICA
CORSO DI INGEGNERIA DEL SOFTWARE

***Gestione del Portale dedicato al
Dottorato di Ricerca in
Informatica***



platform

System Design Document

30/11/2015

1.0

Coordinatori del progetto

<i>Prof. Andrea De Lucia - Top Manager</i>
<i>Elisa D'Eugenio - Project Manager</i>
<i>Dario Di Nucci - Assistant Manager</i>

Partecipanti

<i>Giuseppe Picciocchi - Matricola 0512101820</i>
<i>Tommaso Minichiello - Matricola 0512102296</i>
<i>Matteo Nardone - Matricola 0512102270</i>
<i>Andrea Fedele - Matricola 0512102418</i>
<i>Armando Ariemma - Matricola 0512101778</i>
<i>Michele Santarsiere - Matricola 0512101766</i>

Revision History

Data	Versione	Descrizione	Autori
30/11/2015	1.0	Prima stesura	Team Members
09/01/2016	2.0	Seconda stesura	Andrea Fedele

Indice

1. Introduzione

- 1.1. Proposta del sistema
- 1.2. Obiettivi di progettazione
- 1.3. Definizioni, acronimi, e abbreviazioni
- 1.4. Riferimenti
- 1.5. Panoramica

2. Architettura software corrente

3. Architettura software proposta

- 3.1. Panoramica
- 3.2. Decomposizione in sottosistemi
- 3.3. Mapping hardware/software
- 3.4. Gestione dei dati persistenti
- 3.5. Controllo degli accessi e sicurezza
- 3.6. Controllo software globale
- 3.7. Boundary condition

4. Servizi dei sottosistemi

1 Introduzione

1.1 Proposta del sistema

Lo scopo del sistema è quello di supportare la gestione del portale dedicato al dottorato di ricerca in informatica. Il progetto mira a modificare il Portale già esistente del Dipartimento di Studi e Ricerche Aziendali (Management & Information Technology) dell'Università degli Studi di Salerno riservato ai dottorandi, allo scopo di adattare la piattaforma alle esigenze richieste dai dottorandi. Le modifiche della piattaforma permetteranno di ottimizzare e velocizzare le procedure burocratiche che, nella realtà, rendono difficoltosa l'amministrazione delle proprie esigenze nella gestione dell'anno accademico. Il nuovo sistema dovrebbe permettere ad un dottorando di poter visualizzare la data dei corsi e dei seminari relativi a determinati cicli di dottorato o curriculum. Per rendere più partecipe il dottorando alle attività curriculari il sistema fornirà una gestione delle presenze relative ai corsi ed ai seminari, inoltre riceveranno news e avvisi per quanto riguarda i loro interessi curriculari.

1.2 Obiettivi di progettazione

Il progetto mira a modificare il portale già esistente del Dipartimento di Studi e Ricerche Aziendali (Management & Information Technology) dell'Università degli Studi di Salerno riservato ai dottorandi, allo scopo di adattare il layout grafico alle esigenze degli stakeholder e aggiungere nuove funzionalità richieste dai dottorandi.

Al portale esistente dovranno essere apportate delle modifiche:

- Organizzazione migliore del layout e delle informazioni all'interno del portale;
- Adattamento al database; ci sono informazioni di cui non ha importanza tenere traccia;
- Aggiunta della funzionalità *Calendario*;
- Aggiunta della funzionalità *Presenze*;
- Aggiunta della funzionalità *News & Avvisi*.

Inoltre il sistema dovrà supportare i seguenti criteri:

1. Criteri di performance

- 1.1. Tempi di risposta: i tempi di risposta del sistema dovranno essere brevi, in modo tale da consentire all'utente una navigazione fluida e veloce;
- 1.2. Throughput;
- 1.3. Memoria: la memoria del sistema sarà dinamica in quanto si utilizzerà un database dinamico.

2. Criteri di affidabilità

- 2.1. Robustezza: in caso di errori imprevisti e improvvisi il sistema reagirà bene non causando errori;
 - 2.2. Attendibilità: le pagine dinamiche che produrranno la visualizzazione di informazioni, rispecchieranno istante per istante la reale situazione della base di dati;
 - 2.3. Disponibilità: una volta online, il sistema sarà sempre disponibile a meno di manutenzioni dello stesso;
 - 2.4. Sicurezza: il sistema sarà protetto da un accesso mediante un username e una password.
- 3. Criteri di mantenimento**
- 3.1. Estendibilità: sarà possibile in futuro aggiungere nuove funzionalità al sistema, se richieste;
 - 3.2. Modificabilità: sarà possibile in futuro modificare il sistema;
 - 3.3. Portabilità: il sistema sarà portabile ai vari browser web e quindi su differenti piattaforme;
 - 3.4. Leggibilità: il sistema sarà semplice ed intuitivo all'utente, in modo da consentire una navigazione semplice e intuitiva.
- 4. Criteri End User**
- 4.1. Utilità: l'utente userà il sistema con scioltezza e il sistema supporterà facilmente ogni sua operazione;
 - 4.2. Usabilità: a seconda di chi effettuerà l'accesso al sistema (dottorando, dottore di ricerca, docente o amministratore), il sistema consentirà diverse operazioni.

1.3 Definizioni, acronimi, e abbreviazioni

● Definizioni

- Doc. Ric oppure Ph.D.: dottorando iscritto a un dottorato di ricerca;
- Tutor: docente a cui viene assegnato uno o più dottorandi, per garantire un aiuto e una buona formazione;
- Ciclo: periodo di tempo che dura circa un anno;
- Curriculum: i vari curriculum disponibili per ogni ciclo, che si differenziano per le materie di studio.

● Acronimi

- RAD: Requirements Analysis Document;
- SDD: System Design Document;
- ODD: Object Design Document;
- MU: Manual User;
- DB: Data Base;
- DBMS: Data Base Management System.

● Abbreviazioni

- RF: Requisiti funzionali;
- NRE: Requisiti non funzionali;
- SC: Scenari;
- UC: Casi d'uso;
- UCD: Diagramma dei casi d'uso;
- CD: Class diagram;

- SD: sequence diagram;
- UI: User Interface (mockups).

1.4 Riferimenti

- B. Bruegge, A.H. Dutoit, Object Oriented Software Engineering – Using UML, Patterns and Java, Prentice Hall;
- C. Ghezzi, D. Mandrioli, M. Jazayeri, Ingegneria del Software – Fondamenti e Principi, Prentice Hall;
- I. Sommerville, Software Engineering, Addison Wesley;
- R.S. Pressman, Principi di Ingegneria del Software, Mc Graw Hill;
- Jim Arlow, Ila Neustadt, UML e Unified Process, McGraw-Hill.

1.5 Panoramica

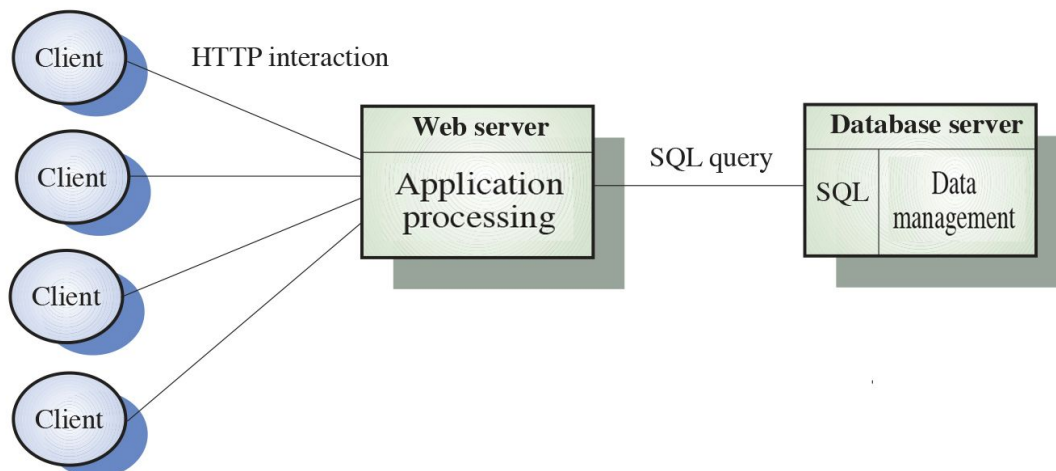
Il documento è stato diviso in quattro parti:

1. Introduzione: riporta una descrizione del sistema corrente e delle modifiche da apportare allo stesso, il motivo per cui viene sviluppato, i vari obiettivi e criteri da raggiungere, una sezione per le definizioni e i riferimenti;
2. Architettura software corrente: riporta la descrizione del sistema corrente;
3. Architettura software proposta: riporta la descrizione del sistema proposto a quello corrente;
4. Servizi dei sottosistemi: riporta una descrizione dei sottosistemi identificati precedentemente e i loro servizi.

2 Architettura software corrente

Il sistema che si vuole sviluppare deriva da una Re-Engineering, in quanto si vogliono riprogettare e quindi re-implementare elementi della piattaforma già esistente.

L'architettura del sistema, però, resta invariata e allo stato attuale essa si presenta come una architettura 3-tier Client/Server come descritto dal seguente schema.



L'architettura del sistema corrente, e quindi anche di quello che sarà il prodotto della Re-Engineering, è descritta in dettaglio nella sezione 3.

3 Architettura software proposta

3.1 Panoramica

Il Sistema commissionato è un software distribuito costituito da un nucleo centrale, contenente i dati e le informazioni relative al contesto universitario. Gli utenti interagiscono con il sistema centrale da diverse postazioni client. I client non hanno alcuna possibilità di interagire tra di loro pertanto, tutte le richieste delle informazioni al database o dei documenti sia in archivio che in locale, dovranno sempre essere fatte al server centrale. Ne deriva che l'architettura più idonea al progetto è di tipo Client-Server. Il sistema pur basandosi su un'architettura Client-Server, permetterà la condivisione di determinati dati come pubblicazioni, missioni, seminari, i quali potranno essere scaricati dal client. Quando un utente richiederà un documento sul portale, la richiesta verrà inviata al server, il quale tramite un'interazione con il DBMS risalirà al documento richiesto, lo stesso verrà prelevato (in copia) dalla postazione e verrà inoltrato all'utente che ne aveva fatto richiesta. Le tipologie di utenti si dividono in: amministratore, studente, docente, utente non registrato. L'amministratore usufruirà della quasi totalità dei servizi offerti dalla piattaforma in quanto rappresenta la figura di amministratore del sistema; lo studente accederà alle funzionalità che si occupano di gestire la carriera universitaria; il docente utilizzerà la piattaforma sia per visionare lo stato di avanzamento della carriera degli studenti sia per comunicare eventuali notizie agli stessi; infine l'utente non registrato potrà prendere visione di informazioni di carattere descrittivo.

3.2 Decomposizione in sottosistemi

Per realizzare il sistema Phd-platform è stato utilizzato lo stile architetturale *three-tier* in versione Client/server. Questa scelta è stata fatta considerando il tipo

di tecnologie utilizzate: infatti, essendo il sistema una piattaforma web, tale stile architetturale risponde esaustivamente a tutte le caratteristiche richieste e rispetta tutti i requisiti non funzionali. L'architettura three-tier ("a tre strati") indica una particolare architettura software di tipo multi-tier per l'esecuzione di un'applicazione web che prevede la suddivisione dell'applicazione in tre diversi moduli o strati dedicati rispettivamente alla interfaccia utente, alla logica funzionale (business logic) e alla gestione dei dati persistenti. Tale architettura va tipicamente a mappare a livello fisico-infrastrutturale quella del sistema informatico ospitante l'applicazione da eseguire.

In particolare i tre strati si occupano di differenti funzionalità del sistema di seguito descritte:

Interface Layer

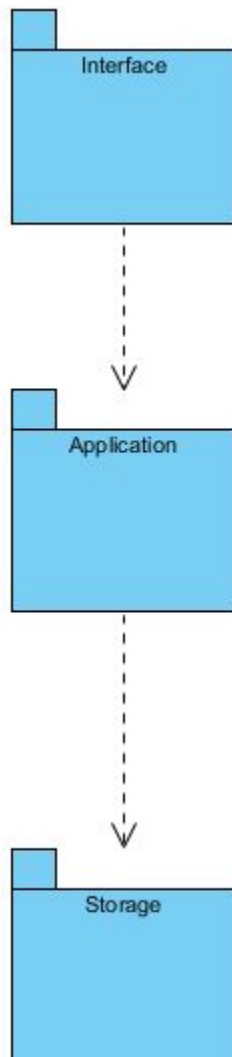
Include tutte le interfacce grafiche e in generale i boundary objects, come le form e i link, con cui interagisce l'utente. L' *interfaccia* verso l'utente è rappresentata da un Web server e da eventuali contenuti statici (es. pagine HTML)

Application Layer

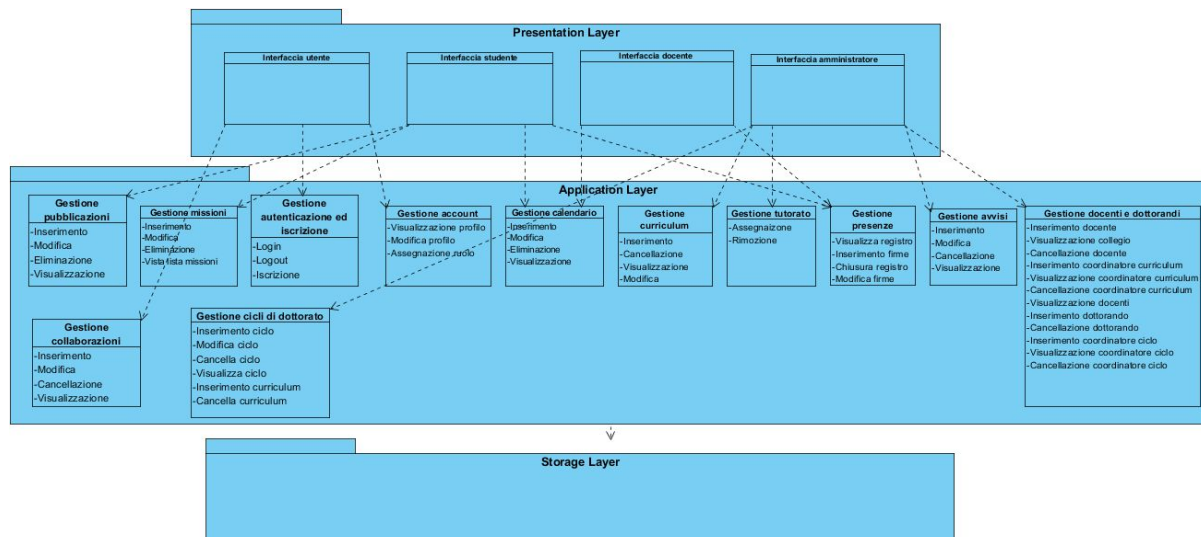
Include tutti gli oggetti relativi al controllo e alle entità del sistema. Tale strato, detto anche *business logic* corrisponde ad una serie di moduli integrati in un application server per la generazione di contenuti dinamici.

Storage Layer

Effettua la memorizzazione, il recupero e l'interrogazione degli oggetti persistenti. I *dati*, i quali possono essere acceduti dalla business logic, sono depositati in maniera persistente su un DBMS (*data layer*). Può risiedere sulla stessa macchina host dell'application server oppure su una macchina host dedicata e separata. Nel sistema tale gestione sarà assegnata all'application server.



Andiamo ora a vedere nello specifico le funzionalità e gli obiettivi di ogni singolo strato implementato nel sistema.



Nello stile architetturale del software phd-platform il **Presentation layer** è caratterizzato dalle interfacce grafiche degli utenti, ovvero dall'interfaccia dell'*utente registrato*, dello *studente*, del *docente* e dell'*amministratore*.

La **Logica applicativa** contiene dodici sottosistemi: ognuno di esso è legato all'interfaccia del sottosistema presente allo strato superiore in base alle funzionalità a cui gli stessi possono accedere.

Lo **Storage Layer** conterrà la base di dati dove verranno resi persistenti i dati, e comunicherà con l'application layer attraverso il framework JDBC.

Il compito del System Design è quello di decomporre il sistema in sottosistemi che mantengano tra di loro un giusto equilibrio tra coesione ed accoppiamento, permettendo così ai team di lavorare sui sottosistemi in modo individuale, con un minimo overhead di comunicazione. La suddivisione nell'application layer è stata fatta seguendo proprio tali criteri. A tal proposito si è deciso di suddividere il sistema in dodici aree, ognuna delle quali si occupa di un determinato aspetto della struttura architetturale. Analizziamo ora i singoli sottosistemi presenti nell'application layer:

Gestione Pubblicazioni

La gestione delle pubblicazioni riguarda la parte che gestisce le pubblicazioni dei dottorandi e permette l'inserimento, la visualizzazione, la modifica e la cancellazione di una pubblicazione, permette anche di visualizzare una lista di tutte le pubblicazioni di un dottorando; Il sistema inoltre permette di inserire, visualizzare, cancellare e scaricare le info di una pubblicazione.

Gestione Missioni

La gestione delle Missioni riguarda la gestione delle missioni di un Dottorando e permette di inserire, modificare e cancellare una missione, inoltre permette di visualizzare una lista delle missioni del Dottorando.

Gestione Autenticazione ed Iscrizione

La gestione dell' Autenticazione ed Iscrizione al portale riguarda la parte che ci permette di gestire il login, il logout e l' iscrizione al portale da parte di un utente.

Gestione Account

La gestione dell' Account riguarda la parte del portale che permette di visualizzare ed apportare le modifiche al proprio account; la possibilità di visualizzare altri account e relative informazioni; la possibilità ad un dottorando di caricare la propria tesi o di scaricare altre tesi di altri dottorandi; la possibilità ad un amministratore di promuovere un utente semplice a dottorando o da dottorando a dottorato.

Gestione Calendario

La gestione dei Calendari riguarda la gestione degli eventi relativi ai corsi e permette di inserire, modificare, cancellare e visualizzare un evento. A sua volta l'evento verrà distinto in lezione e seminario.

Gestione Curriculum

La gestione dei Curriculum riguarda la gestione dei Curriculum di Dottorato e permette l' inserimento, la modifica, la cancellazione e la visualizzazione delle informazioni dei Curriculum di Dottorato.

Gestione Tutorato

La gestione dei Tutorati riguarda la parte di gestione dei tutor dei Dottorandi e permette di assegnare o rimuovere un tutor ad un dottorando.

Gestione Presenze

La gestione delle presenze riguarda la raccolta delle presenze dei dottorandi ai corsi.

Questa gestione permette al docente di aprire una sessione in cui ogni dottorando potrà segnalare la propria presenza e di chiudere tale sessione; inoltre sarà data la possibilità al docente di confermare o no la presenza dei singoli dottorandi.

Gestione Avvisi

La gestione degli avvisi riguarda la gestione degli avvisi legati alla piattaforma ma anche agli avvisi che possono riguardare il proprio Curriculum o seminari o lezione.

Questa gestione permette di inserire, modificare, cancellare e visualizzare un avviso.

Gestione Docente e Dottorandi

La gestione dei Docenti e Dottorandi permette di assegnare ad un Curriculum di un Ciclo i suoi dottorandi, i Docenti, e un Coordinatore , di cancellarli e visualizzarli, inoltre permettere di aggiungere o rimuovere un Docente dal collegio dei Docenti di un ciclo di dottorato.

Gestione Collaborazioni

La gestione delle Collaborazione riguarda la gestione delle collaborazioni di un dottorando e permette l'inserimento, la visualizzazione, la modifica e la cancellazione di una collaborazione.

Gestione Cicli di Dottorato

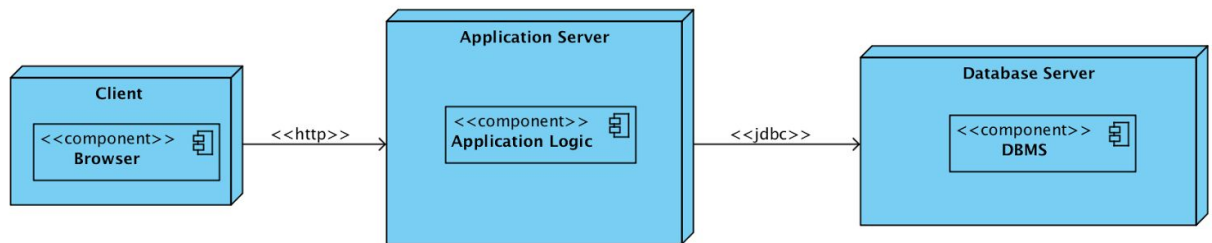
La gestione dei Cicli di Dottorato riguarda la gestione dei cicli di dottorato e permette l'inserimento, la modifica, la visualizzazione e la cancellazione di un ciclo, inoltre permette di collegare un Curriculum ad un Ciclo di dottorato o di rimuovere un Curriculum da un Ciclo.

L'obiettivo principale della *decomposizione in sottosistemi* è ridurre la complessità mentre avvengono cambiamenti. Un sistema grande è di solito decomposto in sottosistemi usando sia gli strati che le partizioni. Il *partizionamento* e la *stratificazione* sono tecniche usate per ottenere una maggiore efficienza dell'intero sistema. In particolare è stata utilizzata la stratificazione per ottenere un Basso Coupling, ovvero che un cambiamento in un sottosistema non incide sugli altri.

3.3 Mapping hardware/software

Molte delle difficoltà della progettazione di un sistema nascono dall'incontro tra l'hardware imposto dall'esterno e i vincoli software.

La struttura hardware proposta è costituita da un server centrale e dalle postazioni utenti che possono essere costituite da computer portatili o fissi nei quali possono girare sistemi operativi diversi fra loro. Al server si collegano i client ed il database, il tipo di utente è determinato in fase di autenticazione controllando nel database il tipo di utente che corrisponde allo Username inserito in quel client. I client dovranno effettuare richieste al server per eseguire le operazioni legate alle loro rispettive funzionalità. Il database conterrà sia i dati relativi agli account, sia i dati relativi ai documenti ed alla loro collocazione e classificazione all'interno del sistema. I client ed il server saranno connesse tramite una rete che utilizzerà il protocollo TCP/IP.

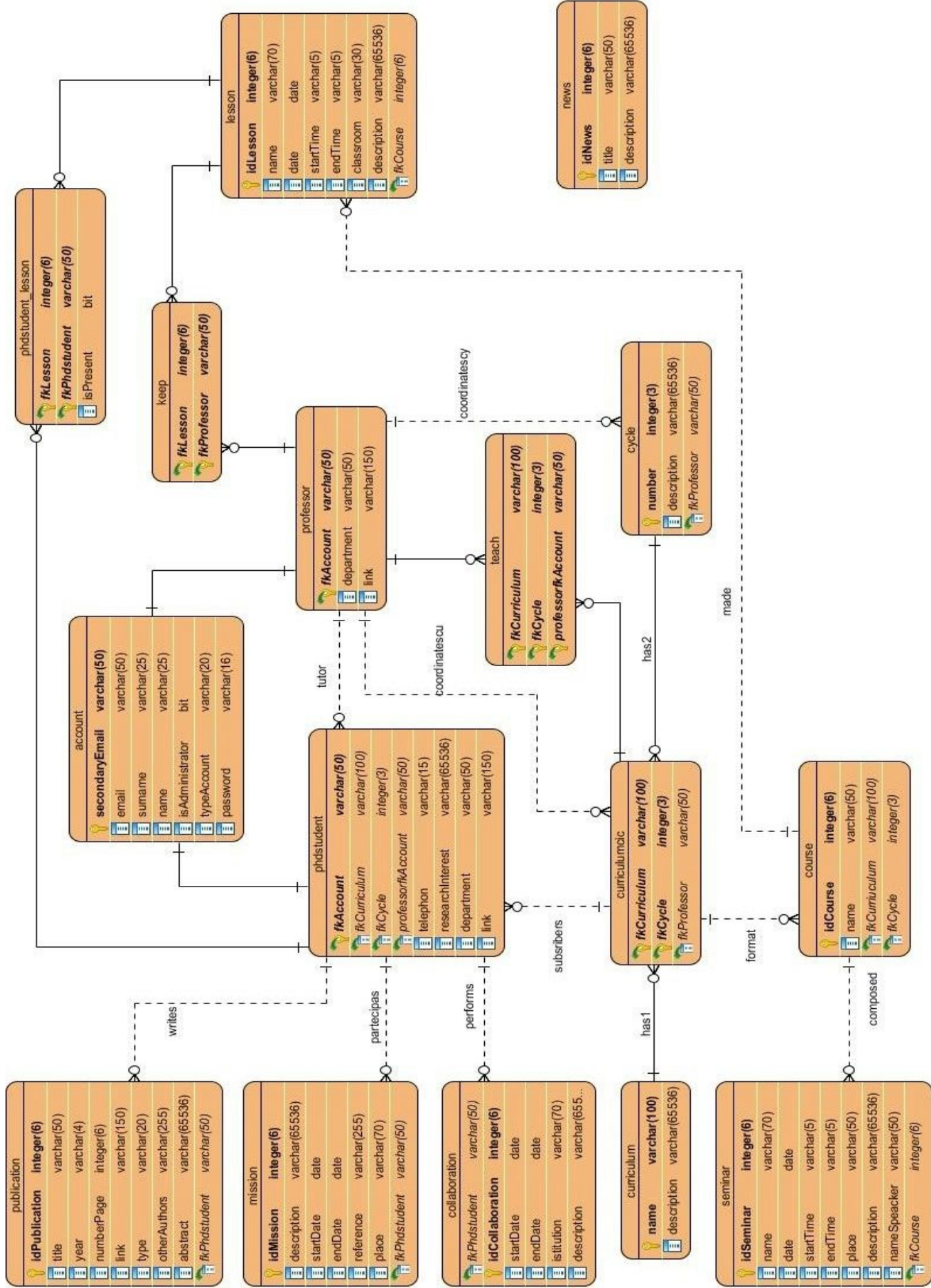


Il sistema Phd-platform come detto, si basa su un'architettura Client-Server. Il Deployment diagram proposto descrive il sistema in termini di risorse hardware e di relazioni tra esse. Come si evince dal diagramma, il server centrale metterà a disposizione dei client la funzionalità di sincronizzazione, che permetterà ai database locali di trasmettere le nuove informazioni acquisite al DBMS centrale e, allo stesso tempo, di percepire eventuali aggiornamenti.

3.4 Gestione dei dati persistenti

3.4.1 Class Diagram entità

Di seguito è riportato il class diagram del portale Phd-platform con un'immagine in verticale per rendere migliore la lettura del diagramma.



3.4.2 Tecnologia Utilizzata

Per la creazione del class diagram delle entità è stato utilizzato il software Visual Paradigm, per la creazione della struttura del database invece è stato utilizzato il software mysql workbench.

è stato deciso di descrivere tutto ciò che riguarda la memorizzazione dei dati permanenti nel documento DBD per rendere facilmente leggibili sia il seguente documento sia l'SDD stesso.

3.5 Controllo degli accessi e Sicurezza

In sistemi multi-utente, attori differenti hanno accesso a funzionalità ed informazioni diverse. Risulta necessario definire una politica di controllo degli accessi e di sicurezza, che identifichi le operazioni ed i privilegi relativi ad ogni attore associato agli elementi presenti all'interno della piattaforma. Da questo si deduce che l'accesso ad ogni elemento di un sistema informatico deve essere regolato e prestabilito per evitare gravi malfunzionamenti ed impedire l'utilizzo di sezioni di sistema riservati ad altri attori. La piattaforma presenta dispositivi di sicurezza e controlli sugli accessi. Essa prevede l'accesso di quattro tipologie di utente: amministratore, utente non accreditato, dottorando e docente. Alla piattaforma è possibile accedere tramite un meccanismo di autenticazione che tramite una form richiede l'inserimento di una Email e di una password. Nel caso l'accesso al sistema non abbia avuto successo la piattaforma mostrerà una schermata di avviso che specifica eventuali errori di inserimento della password o/e di username, consentendo successivamente all'utente di inserire nuovamente le credenziali per effettuare un altro tentativo. Nello specifico la login deve essere l'email di registrazione e la password invece deve essere lunga minimo 8 caratteri. Login e password verranno memorizzate in un database con la particolarità di non essere in chiaro, in quanto il DBMS utilizzerà algoritmi di cifratura per evitare che terze parti riescano ad impossessarsi di dati sensibili. Naturalmente dopo l'autenticazione la piattaforma mostrerà una schermata contenente solo le funzionalità a cui quella tipologia di utente può accedere. Quanto detto prima, però, è escluso per l'utente esterno, che può visualizzare le varie informazioni che non richiedono l'autenticazione.

Per rappresentare in maniera schematica e permettere una lettura immediata delle operazioni consentite agli attori sulle diverse entità, utilizzeremo la lista di controllo degli accessi presentata nel sottoparagrafo successivo.

3.5.1 AM_PUB_1: Gestione Pubblicazioni

<u>Oggetto</u>	Pubblicazione
Attore	
Dottorando	InsertPublication() ModifyPublication() DeletePublication() ViewPublication()
Dottorando Amministratore Utente non accreditato Docente Utente Esterno	ViewPublicationList()

3.5.2 AM_MIS_2: Gestione Missioni

<u>Oggetto</u>	Missione
Attore	
Dottorando	InsertMission() ModifyMission() DeleteMission()
Dottorando Amministratore Utente non accreditato Docente Utente Esterno	ViewMission() ViewMissionList()

3.5.3 AM_ISC_3: Gestione Autenticazione e Iscrizione

<u>Oggetto</u>	Autenticazione e Iscrizione
Attore	
Dottorando Amministratore Utente non Accreditato Docente	Login() Logout()
Utente non Accreditato	SignUp()

3.5.4 AM_ACC_4: Gestione Account

<u>Oggetto</u> Attore	Account
Dottorando Amministratore Utente non Accreditato Docente	ViewProfile() ModifyProfile()
Dottorando	UploadThesis()
Dottorando Amministratore Utente non Accreditato Docente	DownloadThesis()
Amministratore	AssignPrivileges() SendSignUpRequest()
Dottorando Amministratore Utente non Accreditato Docente Utente Esterno	SearchUser()

3.5.5 AM_CAL_5: Gestione Calendario

<u>Oggetto</u> Attore	Calendario
Dottorando Amministratore Utente non Accreditato Docente	ViewEvent()
Docente	InsertEvent() ModifyEvent() DeleteEvent() InsertCourse()

3.5.6 AM_CUR_6: Gestione Curriculum

<u>Oggetto</u> Attore	Curriculum
Amministratore	InsertCurriculum() ModifyCurriculum() DeleteCurriculum()
Dottorando Amministratore Utente non Accreditato Docente Utente Esterno	ViewCurriculum()

3.5.7 AM_TUT_7: Gestione Tutorato

<u>Oggetto</u> Attore	Tutor
Amministratore	AssignTutor() RemoveTutor()

3.5.8 AM_PRE_8: Gestione Presenze

<u>Oggetto</u> Attore	Presenza
Docente	AbilitationSignature() DisableSignature() ViewPresence() CloseRegister()
Dottorando	ViewPresence()

3.5.9 AM_AVV_9: Gestione Avvisi

<u>Oggetto</u>	Avviso
Attore	
Docente	InsertNews() ModifyNews() DeleteNews()
Dottorando Amministratore Utente non Accreditato Docente Utente Esterno	ViewNews() ViewNewsList()

3.5.10 AM_DOC_10: Gestione Docenti e Dottorandi

<u>Oggetto</u>	Docenti e Dottorandi
Attore	
Amministratore	InsertProfessor() ModifyNews() DeleteProfessor() InsertCoordinatorCurriculum() DeleteCoordinatorCurriculum() InsertPhdStudent() DeletePhdStudent() InsertCoordinatorCycle() DeleteCoordinatorCycle()
Dottorando Amministratore Utente non Accreditato Docente Utente Esterno	ViewTeachingStaff() ViewProfessor() ViewPhdStudent() ViewCycle()

3.5.11 AM_COL_11: Gestione Collaborazioni

<u>Oggetto</u> Attore	Collaborazione
Dottorando Amministratore Utente non Accreditato Docente	InsertCollaboration() ModifyCollaboration() DeleteCollaboration()
Dottorando Amministratore Utente non Accreditato Docente Utente Esterno	ViewCollaboration()

3.5.12 AM_CIC_12: Gestione Cicli di Dottorato

<u>Oggetto</u> Attore	Cicli di Dottorato
Amministratore	InsertCycle() DeleteCycle() ModifyCycle() InsertCurriculumToCycle() DeleteCurriculumToCycle()
Dottorando Amministratore Utente non Accreditato Docente Utente Esterno	ViewCycle()

3.6 Controllo Globale del Software

3.6.1 Flusso di controllo esterno

La piattaforma è un sistema interattivo: tutte le funzionalità del sistema vengono avviate solo in seguito ad un comando dell'utente. Per tale motivo, il sistema userà un meccanismo di controllo di flusso di tipo event-driven, grazie all'utilizzo di un linguaggio fortemente event-driven, qual è Java. Il sistema rimane quindi in attesa di una richiesta da parte dell'utente, il quale, tramite la pressione di un bottone scatena un evento, che sarà gestito correttamente dal giusto handler, che

- a sua volta - indirizzerà il controllo del flusso del sistema alla sezione corretta del sottosistema che si occupa della logica di controllo.

3.6.2 Controllo della concorrenza

Più utenti possono accedere contemporaneamente al sistema; è necessario quindi che venga generato un nuovo thread, per ogni utente che richiede un servizio. Sarà poi compito del DBMS occuparsi della concorrenza degli accessi al database. Pertanto bisogna che si valutino bene le condizioni di concorrenza e che vengano prese opportune precauzioni.

3.7 Boundary Condition

3.7.1 Configurazione e start-up del sistema

All'avvio, ai client, si presenta un'interfaccia intuitiva, dalla quale è possibile accedere a un numero limitato di operazioni. Ad autenticazione avvenuta, ogni utente avrà pieno accesso alle funzionalità disponibili per il ruolo che ricopre.

3.7.2 Terminazione del sistema

La terminazione del sistema avviene solo nel caso in cui tutti i sottosistemi siano stati disattivati: nel caso in cui due sottosistemi siano ancora in esecuzione, il sistema rimane attivo. Per non incorrere in problemi - che possano scoraggiare il cliente nell'uso del sistema - prima di disattivare l'application server e il database server, è consigliabile disattivare prima tutti i client. Ogni sottosistema può essere disattivato tramite la funzione di logout, presente su tutte le postazioni, tramite le quali i vari utenti possono accedere al sistema. Questa funzionalità, sulla postazione dell'utente, ha come unico effetto la chiusura del suo terminale; sull'Application server e sul database server, invece, ha l'effetto di disattivare tali sottosistemi.

3.7.3 Fallimento del sistema

Nel caso in cui si verifichi un crash del sistema (dovuto ad un errore hardware o software), si tenta un ripristino della configurazione precedente allo stato d'errore e un riavvio del sistema. Dato che i dati vengono gestiti dal DBMS non c'è rischio di perdite di dati. Tuttavia, in caso di guasti al supporto di memorizzazione dei dati nel database server, inevitabilmente i dati verranno persi. Per rendere minimo tale rischio, verranno eseguiti periodicamente dei backup del database del sistema.

Per evitare problemi è comunque consigliabile effettuare dei controlli periodici sul hardware. In caso di crash dovuto ad un bug nel codice del sistema, si distinguono tre casi:

1. Il crash si è verificato nel client. Il sistema potrà, quindi, continuare a funzionare regolarmente;
2. Il crash si è verificato nel server. Il sistema risulta quindi inutilizzabile, in quanto non è possibile comunicare con il database;
3. Il crash si è verificato nel database server. Il sistema risulta inutilizzabile, in quanto non è possibile accedere ai dati.

4 Servizi dei Sottosistemi

4.0.1 SS_ISC: Gestione Iscrizione e Autenticazione

Application Layer	Presentation Layer	Storage Layer
Login	GUILogin	//
Logout	GUILogin GUILogout	//
Iscrizione	GUILogin GUIIscrizione	//

4.0.2 SS_ACC: Gestione Account

Application Layer	Presentation Layer	Storage Layer
VisualizzaProfili	GUIVisualizzaProfili	//
ModificaProfilo	GUIVisualizzaProfili GUIModificaProfilo	//
AssegnazioneRuoli	GUIAssegnazioneRuoli	//
ConfermaRegistrazioni	GUIConfermaRegistrazioni	//
RicercaAccount	GUIRicercaAccount GUIVisualizzaProfili	//

4.0.3 SS_PUB: Gestione Pubblicazioni

Application Layer	Presentation Layer	Storage Layer
InserisciPubblicazione	GUIInserisciPubblicazione	//
ModificaPubblicazione	GUIVisualizzaPubblicazioni GUIModificaPubblicazione	//
CancellaPubblicazione	GUIVisualizzaPubblicazioni GUICancellaPubblicazione	//
VisualizzaPubblicazioni	GUIVisualizzaPubblicazioni	//

4.0.4 SS_MIS: Gestione Missioni

Application Layer	Presentation Layer	Storage Layer
InserisciMissione	GUIInserisciMissione	//
ModificaMissione	GUIVisualizzaMissioni GUIModificaMissione	//
CancellaMissione	GUIVisualizzaMissioni GUICancellaMissione	//
VisualizzaMissioni	GUIVisualizzaMissioni	//

4.0.5 SS_CAL: Gestione Calendario

Application Layer	Presentation Layer	Storage Layer
InserisciEvento	GUIInserisciEvento	//
ModificaEvento	GUIVisualizzaEventi GUIModificaEvento	//
CancellaEvento	GUIVisualizzaEventi GUICancellaEvento	//
VisualizzaEventi	GUIVisualizzaEventi	//
InserisciCorso	GUIInserisciCorso	//

4.0.6 SS_CUR: Gestione Curriculum

Application Layer	Presentation Layer	Storage Layer
InserisciCurriculum	GUIInserisciCurriculum	//
ModificaCurriculum	GUIVisualizzaCurriculum GUIModificaCurriculum	//
CancellaCurriculum	GUIVisualizzaCurriculum GUICancellaCurriculum	//
VisualizzaCurriculum	GUIVisualizzaCurriculum	//

4.0.7 SS_TUT: Gestione Tutorato

Application Layer	Presentation Layer	Storage Layer
AssegnaTutorato	GUIAssegnaTutorato	//
RimuoviTutorato	GUIRimuoviTutorato	//

4.0.8 SS_PRE: Gestione Presenze

Application Layer	Presentation Layer	Storage Layer
InserisciEvento	GUIInserisciEvento	//
ModificaEvento	GUIVisualizzaEventi GUIModificaEvento	//
CancellaEvento	GUIVisualizzaEventi GUICancellaEvento	//
VisualizzaEventi	GUIVisualizzaEventi	//

4.0.9 SS_AVV: Gestione Avvisi

Application Layer	Presentation Layer	Storage Layer
InserisciAvviso	GUIInserisciAvviso	//
ModificaAvviso	GUIVisualizzaAvvisi GUIModificaAvviso	//
CancellaAvviso	GUIVisualizzaAvvisi GUICancellaAvviso	//
VisualizzaAvvisi	GUIVisualizzaAvvisi	//

4.1.0 SS_DOC: Gestione Docenti e Dottorandi

Application Layer	Presentation Layer	Storage Layer
InserisciDocente	GUIInserisciAvviso	//
VisualizzaCollegio	GUIVisualizzaAvvisi GUIModificaAvviso	//
CancellaDocente	GUIVisualizzaAvvisi GUICancellaAvviso	//
InserisciCoordinatore	GUIVisualizzaAvvisi	//
VisualizzaCoordinatoreCur	GUIVisualizzaCoordinatoreCur	//
CancellaCoordinatoreCur	GUICancellaCoordinatoreCur	//
VisualizzaDocenti	GUIVisualizzaDocenti	//
InserisciDottorando	GUIInserisciDottorando	//
CancellaDottorando	GUICancellaDottorando	//
VisualizzaDottorando	GUIVisualizzaDottorando	//
InserisciCoordinatoreCiclo	GUIInserisciCoordinatoreCiclo	//
VisualizzaCoordinatoreCiclo	GUIVisualizzaCoordinatoreCiclo	//
CancellaCoordinatoreCiclo	GUICancellaCoordinatoreCiclo	//

4.1.1 SS_COL: Gestione Collaborazioni

Application Layer	Presentation Layer	Storage Layer
InserisciCollaborazione	GUIInserisciCollaborazione	//
ModificaCollaborazione	GUIVisualizzaCollaborazioni GUIModificaCollaborazione	//
CancellaCollaborazione	GUIVisualizzaCollaborazioni GUICancellaCollaborazione	//
VisualizzaCollaborazioni	GUIVisualizzaCollaborazioni	//

4.1.2 SS_CIC: Gestione Cicli di Dottorato

Application Layer	Presentation Layer	Storage Layer
InserisciCiclo	GUIInserisciCollaborazione	//
ModificaCiclo	GUIVisualizzaCollaborazioni GUIModificaCollaborazione	//
CancellaCiclo	GUIVisualizzaCollaborazioni GUICancellaCollaborazione	//
VisualizzaCiclo	GUIVisualizzaCollaborazioni	//
InserisciCurriculum	GUIInserisciCurriculum	//
CancellaCurriculum	GUICancellaCurriculum	//