



SDD: System Design Document

RC - Riconoscimento Carriera

Riferimento	
Versione	0.4
Data	03/12/2019
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Agostino Maria Cassese, Andrea Cella, Gerardo Damiano, Vincenzo De Chiara, Giammarco Fonzo, Lorenzo Maturo, Alessandro Quarto, Gianluca Rossi
Approvato da	Domenico Taffuri, Raffaele De Luca



Revision History

Data	Versione	Descrizione	Autori
02/12/2019	0.1	Prima stesura	AQ, AC
03/11/2019	0.2	Aggiunta attributo in esame nel DB	AQ, AC
06/11/2019	0.3	Aggiornamento sul tipo di alcuni attributi nel DB	AQ, AC
12/12/2019	0.4	Aggiornamento database	GD, AC



Sommario

1. Introduzione	4
1.1 Obiettivi del sistema.....	4
1.2 Design Goals	5
1.2.1 Design Trade-off.....	6
1.3 Definizioni, acronimi e abbreviazioni.....	7
1.4 Riferimenti.....	7
1.5 Panoramica.....	7
2. Architettura del Sistema corrente.....	8
3. Architettura del Sistema proposto.....	9
3.1 Panoramica.....	9
3.2 Decomposizione in sottosistemi.....	9
3.2.1 Decomposizione in Layer.....	9
3.2.2 Decomposizione in sottosistemi.....	10
3.2.3 Deployment Diagram.....	12
3.3 Mapping hardware/software	13
3.4 Gestione dati persistenti.....	13
3.5 Controllo degli accessi e sicurezza.....	19
3.6 Controllo flusso globale del sistema	19
3.7 Condizioni limite.....	19
4. Servizi dei sottosistemi	21
5. Glossario	23



1. Introduzione

1.1 Obiettivi del sistema

Il sistema che si vuole realizzare intende migliorare il servizio relativo al riconoscimento della Carriera Pregressa per gli studenti che desiderano richiedere il cambio dell'ateneo, del corso di studi e per gli ex rinunciatari, presso un corso di laurea sia triennale che magistrale in informatica dell'Università degli Studi di Salerno.

L'obiettivo del sistema è dare al consiglio didattico dell'Università degli Studi di Salerno, all'ufficio carriere, ai docenti e agli studenti una piattaforma che consenta loro un processo più rapido ed efficiente, permettendo, inoltre, di gestire meno documentazione cartacea grazie ad un archivio con scopo equivalente ma di gestione digitale.

Per portare a termine questo processo, estenderemo il già presente sistema per la convalidazione di attestati di lingua inglese (English Validation), basandosi anche sui processi burocratici che portano attualmente alla convalida degli esami e dei relativi CFU.

Il nostro sistema verrà incluso nell'attuale web app a cui potranno accedervi lo Studente, l'Ufficio Carriera e il Presidente del Consiglio Didattico, facilitando il trasferimento della documentazione, velocizzando il processo di convalida e diminuendo la possibilità di errori.

Possiamo dividere il sistema in tre categorie:

- Gestione dei file: documentazione caricata dallo studente e report prodotto dal sistema.
- Gestione delle e-mail: e-mail generate dal sistema per facilitare il processo di convalida dei CFU tra PCD e professori.
- Gestione delle notifiche: semplici notifiche mandate agli utenti per informarli sullo stato della richiesta.

Il sistema dovrà consentire il login per tutte e tre le figure individuate (Studente, UC e il PCD) a seconda dei dati di login immessi.

Il sistema dovrà consentire allo Studente di caricare i relativi documenti per iniziare la procedura di convalida, dovrà consentire all'UC la possibilità di scaricare la documentazione, controllarla e successivamente inoltrarla al presidente del consiglio. Il PCD avrà la possibilità tramite un'e-mail auto-generata di inoltrare le richieste ai professori, convalidare il numero di CFU per ogni esame ed inoltrare la richiesta al consiglio didattico.

La nostra estensione dell'attuale sistema, avendo accesso a dati sensibili degli studenti, dovrà attenersi e/o migliorare le attuali metodologie utilizzate, fornendo autenticazioni sicure in modo che i dati di accesso siano protetti.

Il sistema dovrà anche mantenere una facilità di apprendimento e di utilizzo elevata, permettendo una semplice fruizione delle funzionalità offerte, anche senza consultare la documentazione relativa.



1.2 Design Goals

I Design Goals sono organizzati in cinque categorie: Performance, Dependability, Cost, Maintenance, End User Criteria. I Design Goals identificati nel nostro sistema sono i seguenti:

Criteri di performance

- **Tempo di risposta:**

Il prodotto software deve consentire una navigazione rapida ai vari utenti, quindi, tempi di risposta minimi nello svolgimento delle funzionalità da esso offerte.

- **Memoria:**

La memoria necessaria al funzionamento del sistema dipende dalla memoria utilizzata per il mantenimento del Database.

Criteri di affidabilità

- **Robustezza:**

Il sistema informerà l'utente di eventuali errori nel caso di immissione di input non validi attraverso degli appositi messaggi.

- **Affidabilità:**

Il sistema deve garantire l'affidabilità dei servizi proposti. Il prodotto software sarà sviluppato in modo tale da controllare accuratamente le informazioni inserite in input dagli utenti. Il processo di login da parte di tutti gli utenti sarà gestito in modo affidabile, assicurando il corretto funzionamento del sistema.

- **Disponibilità:**

Una volta caricato il sistema sarà disponibile a tutti gli utenti ogni qualvolta gli utenti ne richiederanno l'utilizzo.

- **Tolleranza ai guasti:**

Il sistema potrebbe essere soggetto a fallimenti dovuti a varie cause tra cui un sovraccarico di dati nel database. Per risolvere il problema, è stato previsto un salvataggio automatico dei dati.

- **Sicurezza:**

L'accesso al sistema avviene mediante username e password. Inoltre, la sicurezza è garantita in quanto ogni utente può svolgere solo le operazioni a lui consentite.

Criteri di costi

- **Costo di sviluppo:**

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



Criteri di manutenzione

- Estensibilità:

È possibile aggiungere nuove funzionalità al sistema, dettate dalle esigenze del cliente o dall'avvento di nuove tecnologie.

- Adattabilità:

Il sistema funziona solo per il dipartimento di informatica e per l'Università degli studi di Salerno, ma è adattabile a più dipartimenti o a più università modificando i relativi controlli.

- Portabilità:

L'interazione con il sistema avviene tramite browser, quindi possiamo definirlo portabile poiché il sistema viene sviluppato come una web application, esso è accessibile da qualunque dispositivo, che sia esso mobile o meno. Questa caratteristica garantisce la portabilità dello stesso.

- Tracciabilità dei requisiti:

Attraverso una matrice di tracciabilità, sarà possibile tracciare i requisiti.

Criteri di usabilità

- Usabilità:

Il sistema sarà di facile comprensione e utilizzo, permettendo di effettuare in modo semplice e immediato le varie operazioni grazie a un'interfaccia user-friendly.

- Utilità:

Il sistema si rende utile in quanto velocizza il processo burocratico gestendo i dati in formato digitale e non in maniera cartacea.

1.2.1 Design Trade-off

Memoria vs Estensibilità: Il sistema deve permettere l'estensibilità a discapito della memoria utilizzata. Tale preferenza permette al cliente di richiedere agli sviluppatori nuove funzionalità, dando meno importanza alla memoria utilizzata.

Tempo di risposta vs Affidabilità: Il sistema sarà implementato in modo tale da preferire l'affidabilità al tempo di risposta, in modo tale da garantire un controllo più accurato dei dati in input a discapito del tempo di risposta del sistema.

Disponibilità vs Tolleranza ai guasti: Il sistema deve sempre essere disponibile all'utente in caso di errore in una funzionalità, anche al costo di rendere non disponibile quest'ultima per un lasso di tempo.

Criteri di manutenzione vs Criteri di performance: Il sistema sarà implementato preferendo la manutenibilità alla performance in modo da facilitare gli sviluppatori nel processo di aggiornamento del software a discapito delle performance del sistema.



1.3 Definizioni, acronimi e abbreviazioni

RC: Riconoscimento Carriera;

EV: English Validation;

PCD: Presidente del Consiglio Didattico;

UC: Ufficio Carriera;

CFU: Crediti Formativi Universitari;

RAD: Requirements Analysis Document;

SDD: System Design Document;

USER-FRIENDLY: Letteralmente “amichevole per l’utente”, di facile utilizzo anche per chi non è esperto.

DB: DataBase;

MySQL: È un database Open Source basato sul linguaggio SQL, composto da un client a riga di comando e un server.

SQL: Structured Query Language; linguaggio standardizzato per database basati sul modello relazionale (RDBMS) progettato per: creare e modificare schemi di database.

1.4 Riferimenti

RC_RAD_v1.1

Slide del corso, presenti sulla piattaforma e-learning.

Libro:

- Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition

Autori:

- Bernd Bruegge & Allen H. Dutoit

1.5 Panoramica

Capitolo 1: Contiene l’introduzione con l’obiettivo del sistema, i design goals e un elenco di definizioni, acronimi e abbreviazioni utili alla comprensione dell’intera documentazione.

Capitolo 2: Descrive, nel caso esista, le funzionalità offerte dal sistema corrente.



Capitolo 3: Viene presentata l'architettura del sistema proposto, in cui sarà gestita la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema, le condizioni limite.

Capitolo 4: Vengono presentati i servizi dei sottosistemi.

2. Architettura del Sistema corrente

Attualmente non esiste un sistema digitale per la richiesta di Riconoscimento Carriera pregressa. Per gli studenti che desiderano richiedere il cambio dell'ateneo, infatti i processi per operare questa richiesta vengono effettuati manualmente collegandosi al sito dell'Università e scaricando l'apposito modulo da compilare. Una volta compilato il documento, lo studente dovrà scannerizzarlo insieme al certificato e successivamente inviarli all'ufficio carriere del dipartimento di informatica tramite mail, aspettare che venga approvato dopo essere stato visionato dal presidente del dipartimento e infine dal consiglio didattico. Tutto ciò risulta molto sconveniente per gli studenti, il cui tempo a disposizione è spesso limitato. Per agevolare le operazioni sopra citate, abbiamo deciso di espandere le funzionalità di EV e dare agli utenti la possibilità di usufruire di una piattaforma online anche per il riconoscimento della carriera pregressa. Lo studente potrà svolgere tutte le operazioni da casa, quindi scaricare i documenti e compilarli, inviarli agevolmente e tener sotto controllo lo stato della richiesta tramite semplici notifiche. Le procedure risultano più semplici, sicure e veloci anche per gli amministratori, avendo la possibilità di utilizzare un'unica piattaforma che raccoglie tutte le documentazioni suddividendole in richieste, permettendo anche di tener sotto controllo le diverse fasi e auto-generando le diverse e-mail e report.

3. Architettura del Sistema proposto

3.1 Panoramica

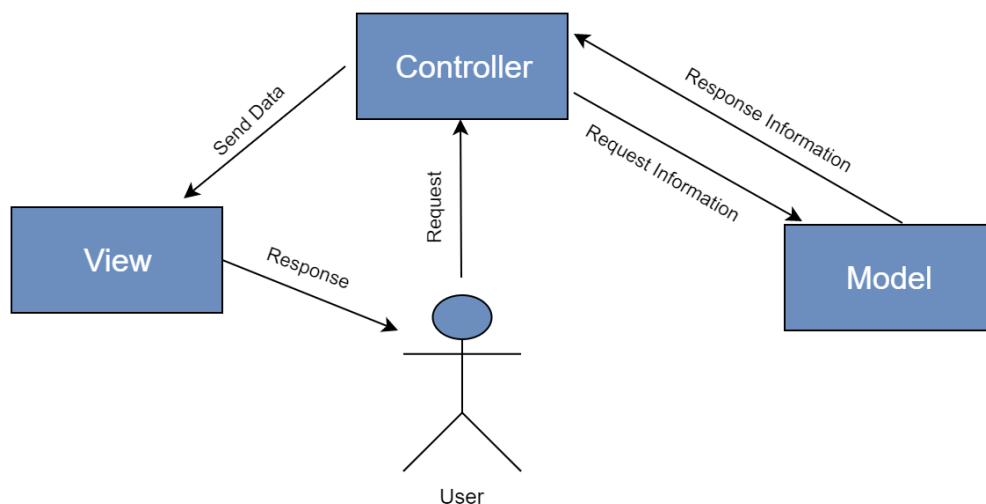
Il sistema da noi proposto è un'applicazione web, che estenderà le funzionalità di English Validation (EV), permettendo di semplificare il processo per il riconoscimento della carriera pregressa. Il sistema prevedrà tre tipi di utenti: studente, ufficio carriere (UC) e presidente consiglio didattico (PCD). Tutti gli utenti potranno effettuare login e log-out; gli studenti avranno la possibilità di registrarsi al sistema. Gli account relativi ad UC e PCD saranno creati durante la fase di installazione del sistema e successivamente forniti agli stessi. Ovviamente, la piattaforma metterà a disposizione diverse tipologie di funzionalità a seconda del tipo di utente che effettuerà l'accesso. In particolare, lo studente potrà creare una nuova richiesta compilando dei form ed allegando due file in formato PDF. Tale richiesta verrà poi inoltrata all'UC, che informato tramite il sistema, provvederà a convalidare la richiesta ed inoltrarla al PCD. Il quale a sua volta potrà inoltrare i programmi di ogni singolo esame, nella richiesta, ai professori di competenza, nel caso in cui il sistema fornisca un matching tra gli esami da valutare e quelli già valutati ed archiviati dal sistema, verrà mostrato un suggerimento relativo al suddetto esame. Il pattern architetturale scelto è di tipo Model-View-Controller (MVC): un pattern architetturale in grado di separare logica di presentazione dei dati, dalla logica di business. È un'architettura multi-tier ovvero le varie funzionalità del sito sono logicamente separate e suddivise su più strati o livelli software differenti in comunicazione tra loro.

3.2 Decomposizione in sottosistemi

3.2.1 Decomposizione in Layer

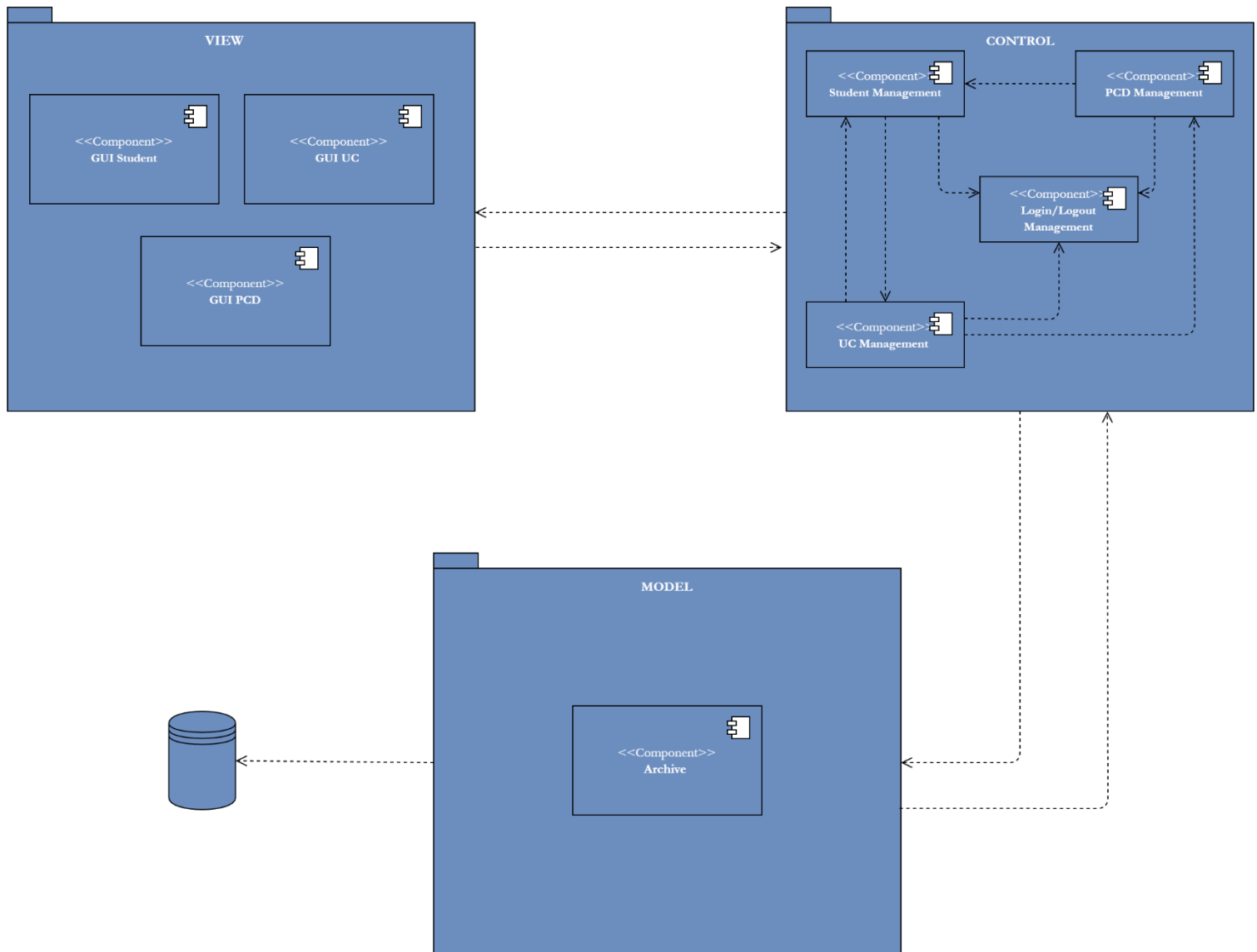
La decomposizione del sistema prevede tre layer che si occupano di gestire aspetti e funzionalità differenti:

- View: raccoglie e gestisce elementi di interfaccia grafica e gli eventi generati su di essi;
- Controller: si occupa della gestione della logica del sistema;
- Model: si occupa della gestione e dello scambio dei dati tra i sottosistemi;



3.2.2 Decomposizione in sottosistemi

Dopo aver analizzato il sistema si è deciso la suddivisione nei seguenti sottosistemi (immagine sottostante), i singoli componenti saranno gestiti con un basso accoppiamento ed una elevata coesione così da non rendere arduo l'implementazione di eventuali modifiche, con un basso accoppiamento sono necessari meno aggiornamenti alle componenti in caso di modifiche.





Il sistema si compone di otto sottosistemi:

Il livello View prevede la gestione di tre sottosistemi:

- GUI Student
- GUI UC
- GUI PCD

Il livello Control prevede la gestione di quattro sottosistemi:

- Login/Logout Management
- Studente Management
- UC Management
- PCD Management

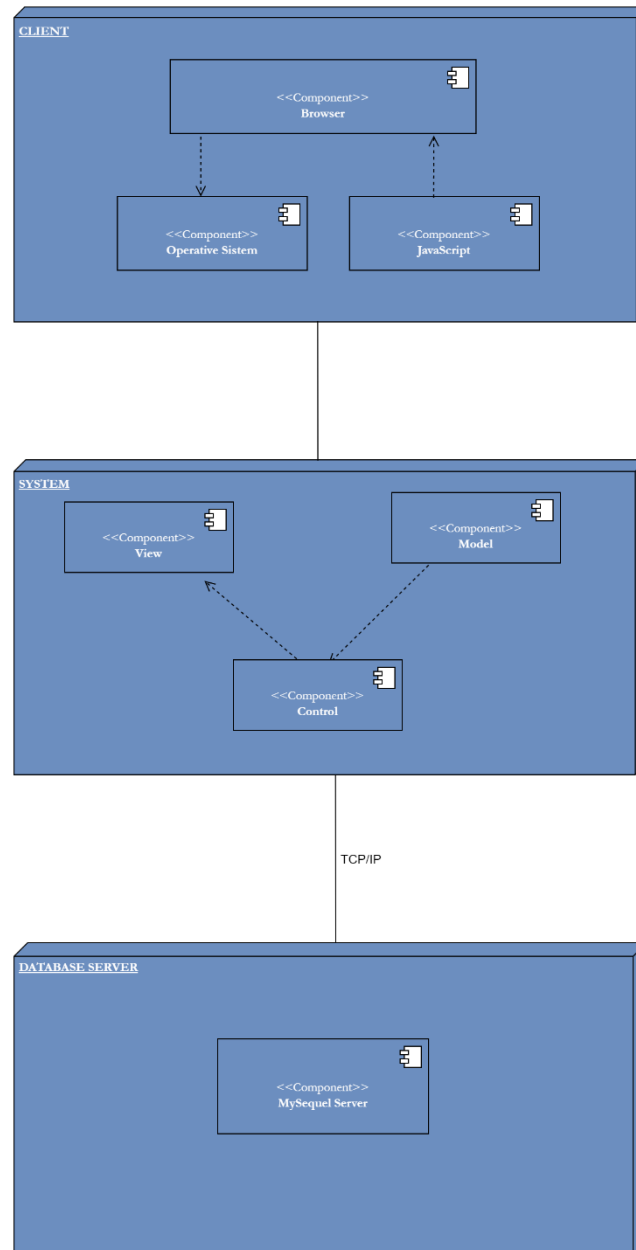
Il livello Model prevede la gestione di un sottosistema:

- Archive

3.2.3 Deployment Diagram

L'utente (Client) richiede le funzionalità tramite l'interfaccia che il sistema mette a disposizione, è però necessario un browser che riesca ad interpretare JavaScript in modo che le funzioni previste dal sistema possano essere eseguite in maniera corretta. L'utente quindi utilizza il livello View del System per raggiungere i suoi obiettivi. La parte Server gestisce e racchiude la persistenza dei dati.

Non sono necessari componenti hardware/software esterni





3.3 Mapping hardware/software

Il sistema utilizzerà una struttura hardware usata come Server che risponderà ai servizi richiesti dai client. Un client può essere rappresentato da qualsiasi macchina attraverso il quale un utente può collegarsi, utilizzando una connessione internet. Il client e il server saranno connessi tramite il protocollo HTTP, con il quale il client inoltrerà delle richieste al server e quest'ultimo provvederà a fornire i servizi richiesti.

Il client necessita solo di una macchina dotata di connessione internet e di un web browser per utilizzare il sistema.

Anche il server necessita di una connessione ad Internet con, inoltre, la capacità di immagazzinare una grande quantità di dati. La componente software necessaria è dunque un DBMS, per consentire la gestione dei dati persistenti.

3.4 Gestione dati persistenti

Per la memorizzazione dei dati si è scelto un Database relazionale che consente un accesso efficiente ai dati, brevi tempi di risposta e un ampio spazio di archiviazione. Inoltre, è garantito l'accesso concorrente ai dati affidabili, ovvero ne viene salvata una copia ed è possibile ripristinare lo stato del database in caso di danni software o hardware. Infine, i dati sono privatizzati, cioè il DBMS ne consente un accesso protetto, quindi utenti diversi con operazioni diverse possono accedere a diverse sezioni del database.

Diagramma E-R ideato per il sistema proposto

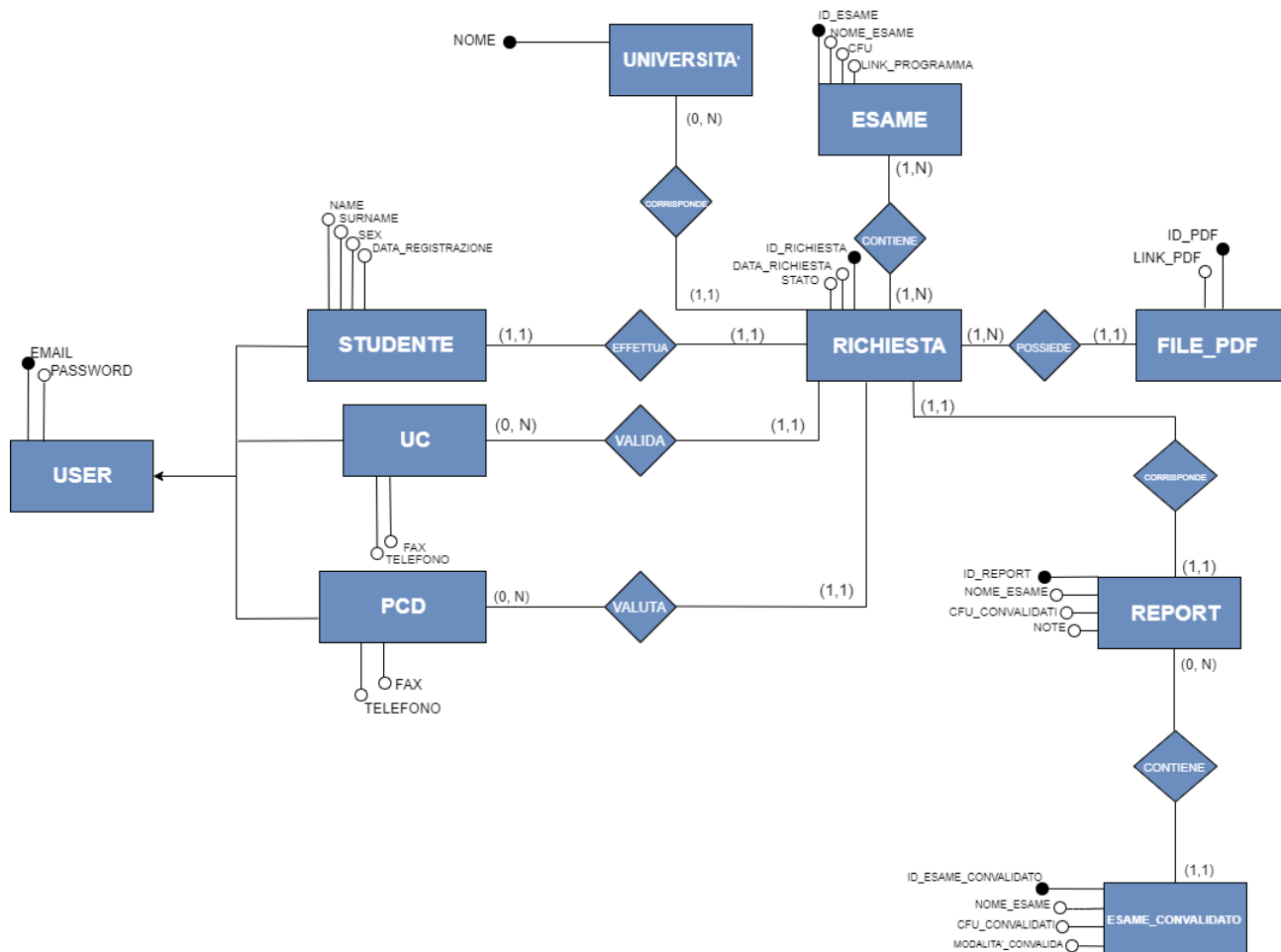
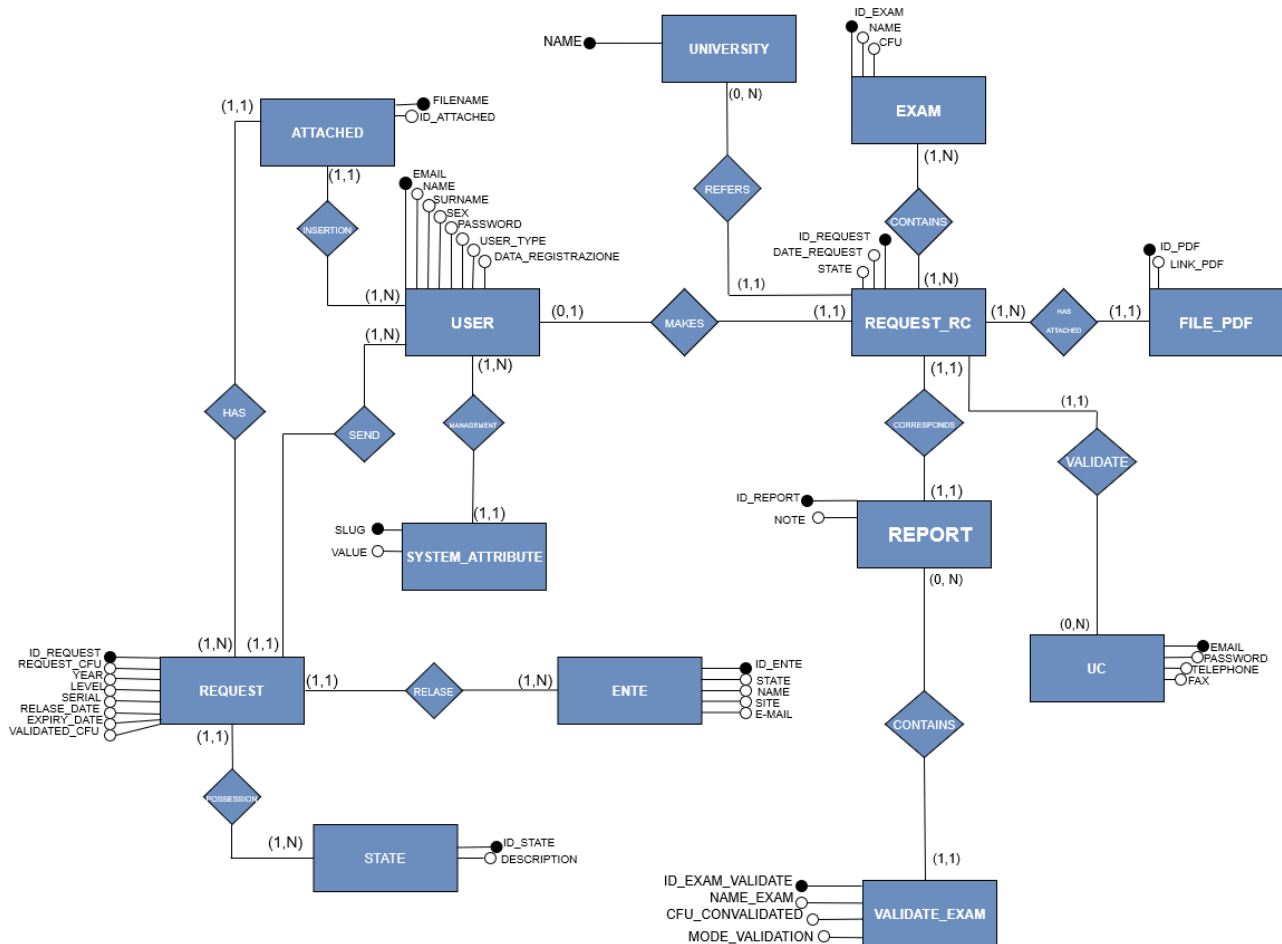
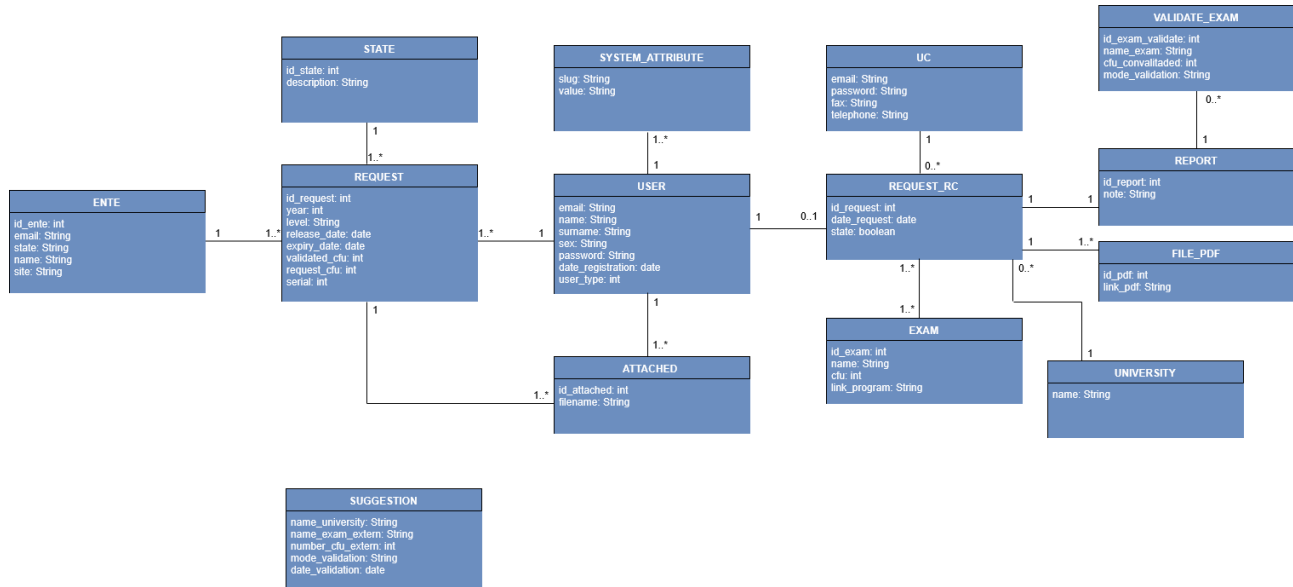


Diagramma E-R ideato per il sistema proposto adattato alla struttura del sistema esistente





Class diagram del database



È stata ideata una entità astratta per implementare la funzionalità relativa ai suggerimenti.

USER

EMAIL NAME SURNAME USER_TYPE SEX PASSWORD DATE_REGISTRATION

REQUEST_RC

ID_REQUEST DATE_REQUEST STATE

EXAM

ID_EXAM NAME CFU LINK_PROGRAM

REPORT

ID_REPORT NOTE



FILE_PDF

ID_PDF LINK_PDF

UC

EMAIL PASSWORD TELEPHONE FAX

VALIDATE_EXAM

ID_EXAM_VALIDATE NAME_EXAM CFU_CONVALIDATED MODE_VALIDATION

UNIVERSITY

NAME

USER

NAME	TYPE	NULL	KEY
EMAIL	VARCHAR (50)	NOT NULL	PRIMARY KEY
NAME	VARCHAR (50)	NOT NULL	
SURNAME	VARCHAR (50)	NOT NULL	
SEX	CHAR	NOT NULL	
PASSWORD	VARCHAR (50)	NOT NULL	
USER_TYPE	TINYINT (1)	NOT NULL	
DATE_REGISTRATION	DATE	NULL	

REQUEST_RC

NAME	TYPE	NULL	KEY
ID_REQUEST	INT(20)	NOT NULL	PRIMARY KEY
DATE_REQUEST	DATE	NOT NULL	
STATE	BOOLEAN	NOT NULL	
FK_UNIVERSITY	VARCHAR (50)	NOT NULL	FOREIGN KEY
FK_USER	VARCHAR (50)	NOT NULL	FOREIGN KEY
FK_REPORT	INT(20)	NOT NULL	FOREIGN KEY
FK_EMAIL_UC	VARCHAR (50)	NOT NULL	FOREIGN KEY



REPORT

NAME	TYPE	NULL	KEY
ID_REPORT	INT(20)	NOT NULL	PRIMARY KEY
NOTE	VARCHAR (5000)	NULL	

VALIDATE_EXAM

NAME	TYPE	NULL	KEY
ID_EXAM_VALIDATE	INT(20)	NOT NULL	PRIMARY KEY
NAME_EXAM	VARCHAR(50)	NOT NULL	
CFU_CONVALIDATED	TINYINT (1)	NOT NULL	
MODE_VALIDATION	VARCHAR(50)	NOT NULL	
FK_ID_REPORT	INT(20)	NOT NULL	FOREIGN KEY

FILE_PDF

NAME	TYPE	NULL	KEY
ID_PDF	INT(20)	NOT NULL	PRIMARY KEY
LINK_PDF	VARCHAR (100)	NOT NULL	
FK_ID_REQUEST_RC	INT(20)	NOT NULL	FOREIGN KEY

EXAM

NAME	TYPE	NULL	KEY
ID_EXAM	INT(20)	NOT NULL	PRIMARY KEY
NAME	VARCHAR (50)	NOT NULL	
CFU	TINYINT (1)	NOT NULL	
LINK_PROGRAM	VARCHAR (256)	NOT NULL	

CONTAINS

NAME	TYPE	NULL	KEY
FK_REQUEST_RC	INT(20)	NOT NULL	PRIMARY KEY
FK_EXAM	INT(20)	NOT NULL	PRIMARY KEY

UNIVERSITY

NAME	TYPE	NULL	KEY
------	------	------	-----



NAME	VARCHAR (100)	NOT NULL	PRIMARY KEY
------	---------------	----------	-------------

UC

NAME	TYPE	NULL	KEY
EMAIL	VARCHAR(50)	NOT NULL	PRIMARY KEY
PASSWORD	VARCHAR(50)	NOT NULL	
TELEPHONE	VARCHAR(10)	NOT NULL	
FAX	VARCHAR(10)	NULL	

È possibile visionare le tabelle del database di English Validation nel relativo documento SDD.

3.5 Controllo degli accessi e sicurezza

In RC sono previsti diversi attori ai quali è concesso di eseguire diverse operazioni. Attraverso l'autenticazione è possibile distinguere il tipo di utente che sta richiedendo l'uso del sistema e in base a questa distinzione è possibile fornire le pagine e le funzioni dedicate allo specifico utente. In questo modo si evita che utenti non autorizzati accedano a funzionalità non destinate a loro.

3.6 Controllo flusso globale del sistema

Il flusso del sistema RC fornisce funzionalità che richiedono continue interazioni con i diversi tipi di utenti (studente, UC, PCD). Per questo motivo il controllo del flusso globale del sistema è di tipo event-driven, ovvero guidato dagli eventi.

3.7 Condizioni limite

Start-up

Per il primo avvio del sistema RC è necessario avviare un web server che fornisca il servizio di un Database MySQL per la gestione dei dati persistenti. Successivamente, tramite l'interfaccia di Login, sarà possibile autenticarsi tramite le opportune credenziali (e-mail, password). Una volta che l'accesso è stato effettuato il sistema presenterà all'utente la pagina iniziale prevista per la sua categoria (le categorie di utenti sono studente, UC, PCD), dalla suddetta pagina sarà possibile usufruire di tutti i servizi che il sistema offre.

Terminazione

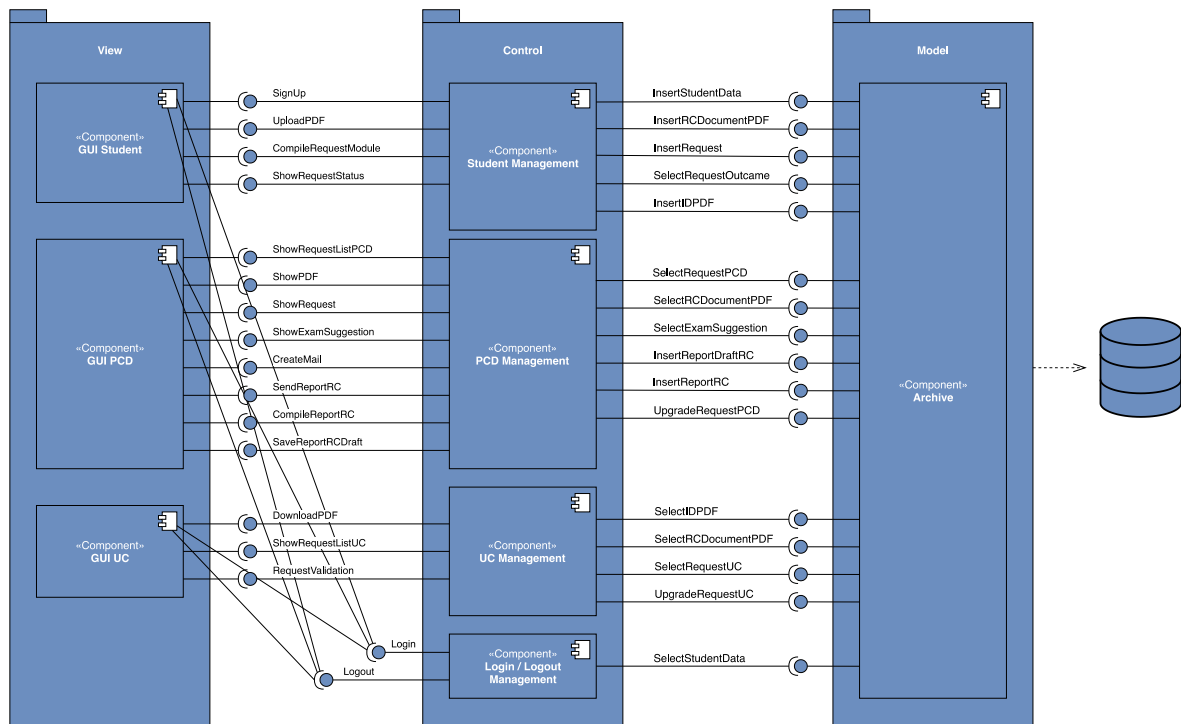
Al momento della corretta chiusura dell'applicazione, si ha la terminazione del sistema con un regolare log-out. Per consentire la corretta terminazione del server, l'amministratore del sistema dovrà effettuare la procedura di terminazione, dopo la quale nessun client potrà connettersi al sistema.

Fallimento



1. Nel caso di guasti dovuti al sovraccarico del database con successivo fallimento dello stesso è prevista come procedura preventiva il salvataggio periodico dei dati sotto forma di codice SQL per la successiva rigenerazione del DB.
2. Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione non sono previsti metodi che ripristino lo stato del Sistema precedente allo spegnimento non voluto.
3. Un altro caso di fallimento potrebbe derivare dal software stesso che causa una chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione. Non essendo previste politiche correttive, l'unica operazione consentita in questa particolare situazione è la chiusura del sistema e il suo successivo riavvio.
4. Un altro caso di fallimento potrebbe essere dovuto ad un errore critico nell'hardware, contro il quale non è prevista alcuna contromisura.

4. Servizi dei sottosistemi



Student Management offre 4 servizi all'interfaccia View:

- SignUp;
- UploadPDF;
- CompileRequestModule;
- ShowRequestStatus.

PCD Management offre 8 servizi all'interfaccia View:

- ShowRequestListPCD;
- ShowPDF;
- ShowRequest;
- ShowExamSuggestion;
- CreateMail;
- SendReportRC;
- CompileReportRC;



- SaveReportRCDraft.

UC Management offre 3 servizi all'interfaccia View:

- DownloadPDF;
- ShowRequestListUC;
- RequestValidation.

Login / Logout Management offre 2 servizi all'interfaccia View:

- Login;
- Logout;

Archive offre 16 servizi all'interfaccia Control:

- InsertStudentData;
- InsertRCDocumentPDF;
- InsertIDPDF;
- InsertRequest;
- SelectRequestOutcome.
- SelectRequestPCD;
- SelectRCDocumentPDF;
- SelectExamSuggestions;
- InsertReportDraftRC;
- InsertReportRC;
- UpgradeRequestPCD.
- SelectIDPDF;
- SelectRCDocumentPDF;
- SelectRequestUC;
- UpgradeRequestUC.
- SelectStudentData.



5. Glossario

RC: Riconoscimento Carriera;

EV: English Validation;

PCD: Presidente del Consiglio Didattico;

UC: Ufficio Carriera;

CFU: Crediti Formativi Universitari;

RAD: Requirements Analysis Document;

SDD: System Design Document;

USER-FRIENDLY: Letteralmente “amichevole per l’utente”, di facile utilizzo anche per chi non è esperto.

DB: DataBase;

MySQL: È un database Open Source basato sul linguaggio SQL, composto da un client a riga di comando e un server.

SQL: Structured Query Language; linguaggio standardizzato per database basati sul modello relazionale (RDBMS) progettato per: creare e modificare schemi di database.