

Unistudy

Object Design Document

Repository:
FrancescoScala/UniStudy

Ingegneria del Software
2022/2023

Università degli Studi di
Salerno



PARTECIPANTI AL PROGETTO

NOME	MATRICOLA
Mauro Nicola	0512110047
Scala Francesco	0512110509

Scritto da:	Nicola Mauro Francesco Scala
-------------	---------------------------------

Revision history

Data	Versione	Descrizione	Autore
19/12/2022	1.0	Stesura iniziale Object Design Document e individuazione Trade-offs	F.S. N.M.
20/12/2022	1.1	Inserimento Class Diagram	N.M.
22/12/2022	1.2	Definizione contratti OCL MemberManager	F.S.
22/12/2022	1.2	Definizione contratti OCL CourseManager	N.M.
23/12/2022	1.3	Definizione contratti OCL NoticeManager	F.S.

23/12/2022	1.3	Definizione contratti OCL NoteManager	N.M.
23/12/2022	1.3	Definizione contratti OCL EnrollmentManager	F.S. N.M.
18/02/2023	2.0	Rivista "Organizzazione delle directory"	F.S. N.M.
11/03/2023	2.1	Aggiunto nuovo Class Diagram	F.S.
15/03/2023	2.2	Revisione finale	F.S. N.M.

Indice

Overview	5
Trade-offs	5
Linee guida per la documentazione delle interfacce	5
Organizzazione delle directory.....	6
Glossario delle interfacce di classe	8
Contratti OCL	9

Overview

Il documento in esame espone la progettazione dettagliata del sistema. Dopo aver presentato i principali trade-offs di progettazione ed una breve descrizione delle linee guida adoperate per documentare le interfacce dei moduli, verrà illustrata la specifica delle stesse, e l'organizzazione delle classi implementate.

Trade-offs

Seguono i trade-offs che hanno guidato lo sviluppo dell'applicativo:

1. Tempo di risposta VS Spazio occupato in sessione

Al fine di diminuire il tempo di risposta del caricamento delle pagine del sistema, si è optato per la memorizzazione in sessione dei dati dell'utente e delle sue iscrizioni (enrollment) all'atto del login.

2. Sviluppo rapido VS Funzionalità

Per velocizzare il rilascio del prodotto si è scelto di optare in questa prima release per l'implementazione delle funzionalità con priorità maggiore. Le rimanenti saranno oggetto di sviluppi successivi.

Linee guida per la documentazione delle interfacce

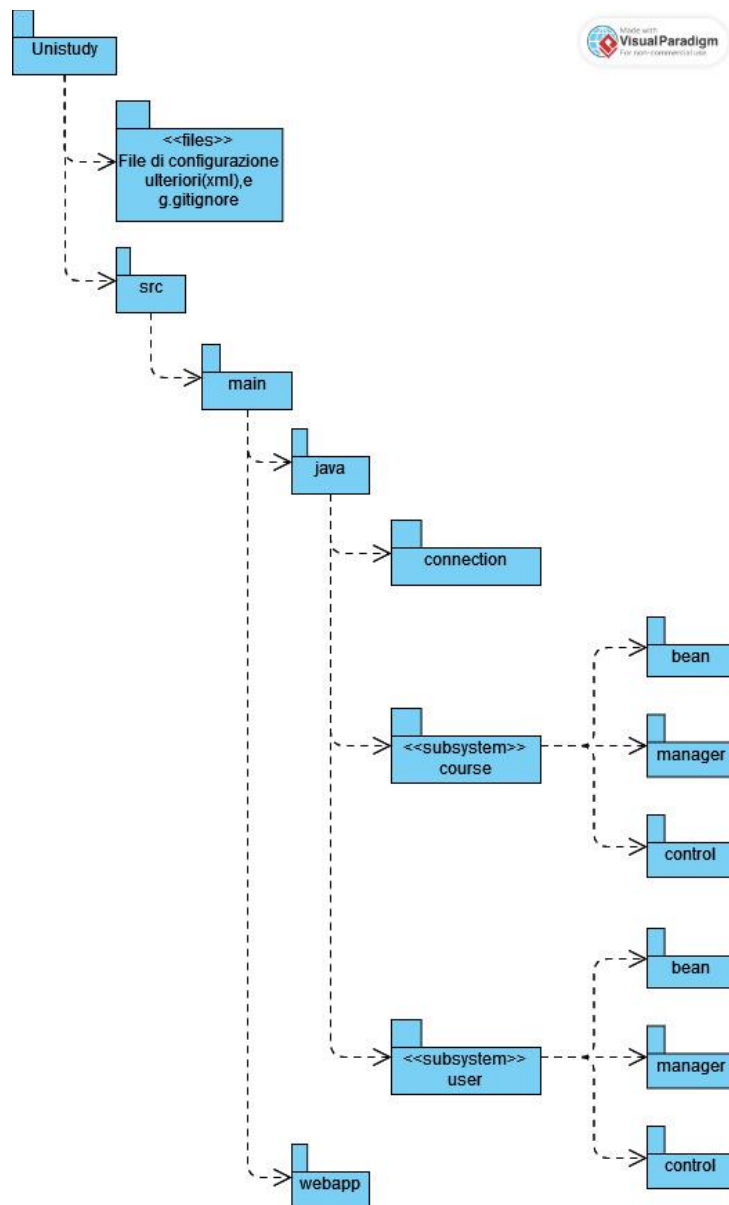
Le interfacce dei moduli implementati saranno soggette ai criteri seguenti:

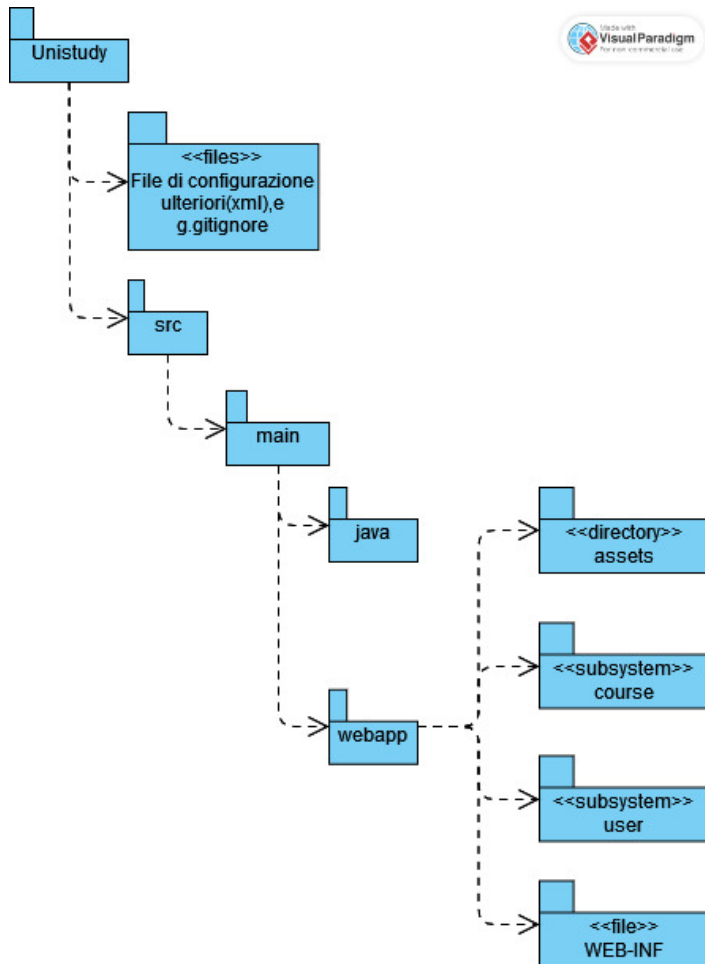
1. Le classi sono identificate da nomi singolari;
2. I nomi degli attributi e dei metodi seguiranno la convenzione "camel-case";
3. Tutti i metodi getters/setters correlati ad un attributo saranno esplicitati con il nome dell'attributo che segue get/set (es. get...()/set...());
4. Eventuali errori verranno gestiti attraverso le eccezioni, senza alcun valore di ritorno;
5. I nomi delle classi Control e Manager saranno preceduti dal nome della classe di riferimento;
6. Precondizioni, post condizioni ed invarianti di classe saranno esplicitati mediante OCL;
7. I nomi delle operazioni seguono lo standard "entity:operation";

Organizzazione delle directory

Ogni sottosistema identificato nel SDD verrà mappato su una directory ben precisa. Nel seguente package diagram, ogni package è una directory del progetto, mentre gli archi rappresentano la relazione "include". Per facilitarne la comprensione si è deciso di riportare separatamente il back e il front end del sistema.

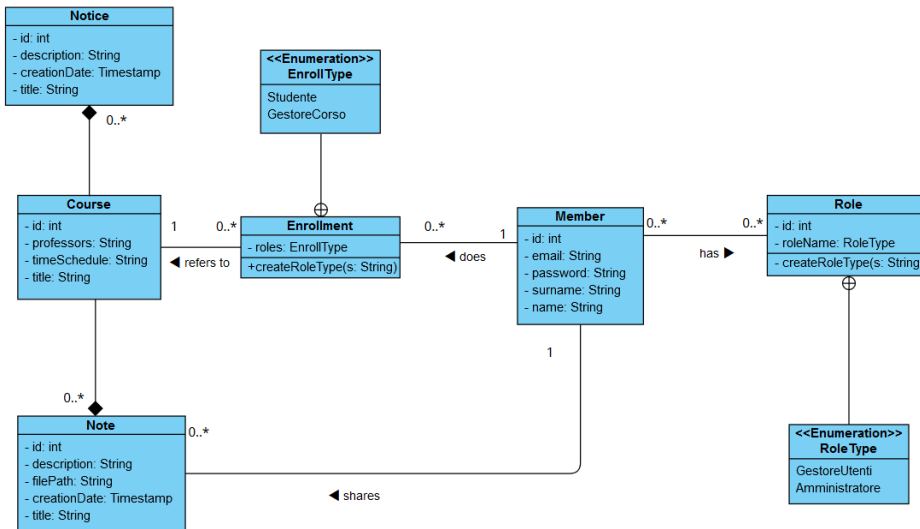
LATO BACK-END



LATO FRONT-END

Glossario delle interfacce di classe

Segue una modellazione più completa del Class Diagram presentato all'interno del RAD.



Contratti OCL

Classe MemberManager:

Context MemberManager::signupMember(email, password, name, surname) pre: <ul style="list-style-type: none"> – email != null – password !=null – name != null – surname != null Context MemberManager:: signupMember(email, password, name, surname) post: <ul style="list-style-type: none"> – true
Context MemberManager::loginMember(email, password) pre: <ul style="list-style-type: none"> – email != null – password !=null Context MemberManager:: loginMember(email, password) post: <ul style="list-style-type: none"> – member.getId() != null – member.getEmail() = email – member.getPassword() = password – member.getName() != null – member.getSurname() != null
Context MemberManager::modifyInfoMember(member,name,surname,newPassword,oldPassword) pre: <ul style="list-style-type: none"> – member!= null – name != null – surname != null – newPassword !=null – oldPassword != null Context MemberManager::modifyInfoMember(member,name,surname,newPassword,oldPassword) post: <ul style="list-style-type: none"> – true
Context MemberManager:: retrieveIdMemberByEmail(email) pre: <ul style="list-style-type: none"> – email != null Context MemberManager:: retrieveIdMemberByEmail(email) post: <ul style="list-style-type: none"> – id != null

Classe CourseManager

Context CourseManager:: createCourse(professors, schedule, title) pre: <ul style="list-style-type: none"> – professors != null – schedule != null – title !=null Context CourseManager:: createCourse(professors, schedule, title) post: <ul style="list-style-type: none"> – true
Context CourseManager:: modifyInfoCourse(course, newProfessors, newSchedule) pre: <ul style="list-style-type: none"> – course !=null

<ul style="list-style-type: none"> – newProfessors != null – newSchedule != null Context CourseManager:: modifyInfoCourse(course, newProfessors, newSchedule) post: <ul style="list-style-type: none"> – true
Context CourseManager:: retrieveCourseById(courseId) pre: <ul style="list-style-type: none"> – courseId !=null Context CourseManager:: retrieveCourseById(courseId) post: <ul style="list-style-type: none"> – course !=null – course.getId() = courseId – course.getTitle() != null – course.getProfessors() != null – course.getTimeSchedule() != null
Context CourseManager:: retrieveIdCourseByTitle(courseTitle) pre: <ul style="list-style-type: none"> – courseTitle!=null Context CourseManager:: retrieveIdCourseByTitle(courseTitle) post: <ul style="list-style-type: none"> – id!=null
Context CourseManager:: retrieveAll() pre: <ul style="list-style-type: none"> – true Context CourseManager:: retrieveAll() post: <ul style="list-style-type: none"> – courses != null – self->forAll(c: Course c.getId() = null ^ course.getId() = courseId ^ course.getTitle() != null ^ course.getProfessors() != null ^ course.getTimeSchedule() != null)
Context CourseManager:: deleteCourse(id) pre: <ul style="list-style-type: none"> – id!=null Context CourseManager:: deleteCourse(id) post: <ul style="list-style-type: none"> – true

Classe EnrollmentManager

Context Enrollmentmanager:: createEnrollment(memberId, courseId, enrollType, courseTitle) pre: <ul style="list-style-type: none"> – memberId != null – courseId != null – enrollType != null – courseTitle !=null Context Enrollmentmanager:: createEnrollment(memberId, courseId, enrollType, courseTitle) post: <ul style="list-style-type: none"> – enrollment.getMemberId() = memberId – enrollment.getCourseId() = courseId – enrollment.getCourseTitle() = courseTitle – enrollment.getRoles().size=1 ^ enrollment.getRoles()->includes(enrollType)
Context Enrollmentmanager:: retirieveEnrollmentsByMemberId(memberId) pre: <ul style="list-style-type: none"> – memberId != null

Context Enrollmentmanager:: retirieveEnrollmentsByMemberId(memberId) post:
<ul style="list-style-type: none"> – enrollments != null – self->forAll(e: Enrollment e.getMemberId() = memberId ^ e.getCourseld() != null ^ e.getCourseTitle() != null ^ e.getRoles() != null)
Context Enrollmentmanager:: deleteEnrollment(memberId, courseld) pre:
<ul style="list-style-type: none"> – memberId != null – courseld != null
Context Enrollmentmanager:: deleteEnrollment(memberId, courseld) post:
<ul style="list-style-type: none"> – true
Context Enrollmentmanager:: updateEnrollment(enrollment, type) pre:
<ul style="list-style-type: none"> – enrollment != null – type != null
Context Enrollmentmanager:: updateEnrollment(enrollment, type) post:
<ul style="list-style-type: none"> – enrollment.getRoles()->includes(type)
Context EnrollmentManager :: unenroll(enrollType, e) pre:
<ul style="list-style-type: none"> – e != null – enrollType != null
Context EnrollmentManager :: unenroll(enrollType, e) post:
<ul style="list-style-type: none"> – true
Context Enrollmentmanager:: removeRoleEnrollment(memberId, courseld, enrollType) pre:
<ul style="list-style-type: none"> – memberId != null – courseld != null – enrollType != null
Context Enrollmentmanager:: removeRoleEnrollment(memberId, courseld, enrollType) post:
<ul style="list-style-type: none"> – true

Classe NoteManager

ContextNoteManager::createNote(description,creationDate,filepath,title,authorId,authorInfo,courseld) pre:
<ul style="list-style-type: none"> – description != null – creationDate != null – filepath != null – authorId !=null – title != null – authorInfo != null – courseld != null
ContextNoteManager::createNote(description,creationDate,filepath,title,authorId,authorInfo,courseld) post:
<ul style="list-style-type: none"> – true
Context NoteManager :: retreiveNotesByCourseld(courseld) pre:
<ul style="list-style-type: none"> – courseld != null
Context NoteManager :: retreiveNotesByCourseld(courseld) post:

<ul style="list-style-type: none"> – notes != null – self->forall(n: Note n.getCourseld() = courseld ^ n.getDescription() != null ^ n.getCreationDate() != null ^ n.getFilepath() != null ^ n.getAuthorId() != null ^ n.getTitle() != null ^ n.getAuthorInfo() != null)
Context NoteManager :: deleteNote(id) pre: <ul style="list-style-type: none"> – id != null Context NoteManager :: deleteNote(id) post: <ul style="list-style-type: none"> – true

Classe NoticeManager

Context NoticeManager :: createNotice(title, creationDate, description, courseld) pre: <ul style="list-style-type: none"> – title != null – creationDate != null – description != null – courseld != null Context NoticeManager :: createNotice(title, creationDate, description, courseld) post: <ul style="list-style-type: none"> – true
Context NoticeManager :: retrieveNoticesByCourseld(courseld) pre: <ul style="list-style-type: none"> – courseld != null Context NoticeManager :: retrieveNoticesByCourseld(courseld) post: <ul style="list-style-type: none"> – notices != null – self->forall(n: Notice n.getCourseld() = courseld ^ n.getDescription() != null ^ n.getCreationDate() != null ^ n.getTitle() != null)
Context NoticeManager :: deleteNotice(id) pre: <ul style="list-style-type: none"> – id != null Context NoticeManager :: deleteNotice(id) post: <ul style="list-style-type: none"> – true