OCL

Object Constraint Language

OCL CONSTRAINTS

-- Student Constraints

context Student inv YearOfGraduationNotEmpty:

not self.yearOfGraduation.oclIsUndefined() and self.yearOfGraduation <> ''

context Student inv OneApplicationPerInternship:

self.user.applications->isUnique(a | a.internship)

--------------------------------------------------------------------------------------------------------------------------------------------------------

-- Application Constraints

context Application inv MustHaveStudentAndInternship:

not self.studentId.oclIsUndefined() and not self.internship.oclIsUndefined()

context Application inv CreatedBeforeOrAtUpdated:

self.createdAt <= self.updatedAt

----------------------------------------------------------------------------

-- Internship Constraints

context Internship inv ValidInternshipType:

InternshipType->includes(self.internshipType)

context Internship inv ValidStatus:

Status->includes(self.status)

--------------------------------------------------------------------------------------------------------------------------------------------------------

-- HR Constraints

context Hr inv MustBeLinkedToCompanyAndUser:

not self.company.oclIsUndefined() and not self.user.oclIsUndefined()

--------------------------------------------------------------------------------------------------------------------------------------------------------

-- Notification Constraints

context Notification inv MustHaveStudentAndApplication:

not self.studentId.oclIsUndefined() and not self.application.oclIsUndefined()

--------------------------------------------------------------------------------------------------------------------------------------------------------

-- Admin Constraints

context Admin inv MustBeLinkedToUser:

not self.user.oclIsUndefined()

--------------------------------------------------------------------------------------------------------------------------------------------------------

PRE/POST CONDITIONS

-- submitApplication

context Student::submitApplication(internship: Internship)

pre:

not self.user.oclIsUndefined() and

internship.status = Status::ACTIVE and

self.user.applications->select(a | a.internship = internship)->isEmpty()

post:

self.user.applications->exists(a | a.internship = internship and a.studentId = self.user.id)

--------------------------------------------------------------------------------------------------------------------------------------------------------

-- updateApplicationStatus

context Hr::updateApplicationStatus(application: Application, newStatus: Status)

pre:

self.company.name = application.internship.companyName

post:

application.status = newStatus

--------------------------------------------------------------------------------------------------------------------------------------------------------

-- publishInternship

context Hr::publishInternship(internship: Internship)

pre:

internship.title <> '' and

internship.location <> '' and

not internship.internshipType.oclIsUndefined() and

internship.duration <> ''

post:

internship.status = Status::ACTIVE

----------------------------------------------------------------------------

-- notifyStudent

context Hr::notifyStudent(application: Application)

pre:

not application.studentId.oclIsUndefined() and

not application.internship.oclIsUndefined()

post:

application.studentId.notifications->exists(n | n.application = application and n.isRead = false)

--------------------------------------------------------------------------------------------------------------------------------------------------------

-- registerAccount

context System::registerAccount(user: User)

pre:

user.username <> '' and user.password <> '' and user.email <> ''

post:

User.allInstances()->includes(user)

--------------------------------------------------------------------------------------------------------------------------------------------------------

-- validateResume

context System::validateResume(student: Student)

pre:

not student.cv.oclIsUndefined() and student.cv <> ''

post:

student.cv.matches('[a-zA-Z0-9:/\_.-]+')

--------------------------------------------------------------------------------------------------------------------------------------------------------

-- autoCloseInternships

context System::autoCloseInternships()

pre:

Internship.allInstances()->exists(i | i.status = Status::ACTIVE)

post:

Internship.allInstances()->forAll(i |

i.status = Status::EXPIRED implies i.createdAt.addMonths(3) <= now()

)