Changelog

1. Changes in UML domain model

Application

- The credit card relation on application has been deleted. A new credit card attribute has been added.

CreditCard

- CreditCard class has been changed to Datatype due to how it was before the same card could be repeated several times in a user. Being a datatype, a user can only add one to his application or to his sponsorship.

Complaint

- @URL annotation has been deleted from Complaint::attachments. This annotation is not compatible with multiple URL in the same attribute.

Customisation

- Customisation::languages has been added as attribute.

EducationRecord

- @URL annotation has been deleted from EducationRecord::attachment. This annotation is not compatible with multiple URL in the same attribute.

MiscellaneousRecord

- @URL annotation has been deleted from MiscellaneousRecord::attachment. This annotation is not compatible with multiple URL in the same attribute.

ProfessionalRecord

- @URL annotation has been deleted from ProfessionalRecord::attachment. This annotation is not compatible with multiple URL in the same attribute.

Report

- @URL annotation has been deleted from Report::attachments. This annotation is not compatible with multiple URL in the same attribute.

Sponsorship

- The credit card relation on sponsorship has been deleted. A new credit card attribute has been added.

2. Changes in JAVA domain model

Application

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy HH:mm") at Application::registerMoment getter.
- In the attribute creditCard the annotation oneToMany has been deleted.

CreditCard

- The annotation @Entity has been changed by @Embeddable to be a datatype.

Complaint

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy HH:mm") at Complaint::moment getter.
- @URL annotation has been deleted from Complaint::attachments. This annotation is not compatible with multiple URL in the same attribute.

Education Record

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy") at EducationRecord::startDate getter.
- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy") at EducationRecord::endDate getter.
- @URL annotation has been deleted from EducationRecord::attachment. This annotation is not compatible with multiple URL in the same attribute.

Endorserment

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy HH:mm") at Endorsement::moment getter.

Finder

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy ") at Finder::startDate getter.
- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy ") at Finder::endDate getter.
- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy HH:mm ") and @Temporal(TemporalType.TIMESTAMP) at Finder::lastUpdate getter.

FixUpTask

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy HH:mm") at FixUpTask::publicationMoment getter.
- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy") at FixUpTask::startDate getter.
- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy") at FixUpTask::endDate getter.

Message

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy HH:mm") at Message::sendMoment getter.

MiscellaneousRecord

- @URL annotation has been deleted from MiscellaneousRecord::attachment. This annotation is not compatible with multiple URL in the same attribute.

Note

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy HH:mm") at Note::moment getter.

Phase

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy HH:mm") at Phase::startMoment getter.
- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy HH:mm") at Phase::endMoment getter.

Professional Record

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy ") at ProfessionalRecord::startDate getter.
- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy") at ProfessionalRecord::endDate getter.
- @URL annotation has been deleted from ProfessionalRecord::attachment. This annotation is not compatible with multiple URL in the same attribute.

Report

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy HH:mm") at Report::moment getter.
- @URL annotation has been deleted from Report::attachments. This annotation is not compatible with multiple URL in the same attribute.

Sponsorship

- In the attribute creditCard the annotation oneToMany has been deleted.

Tutorial

- Added annotation @DateTimeFormat(pattern = "dd/MM/yyyy HH:mm") at Tutorial::moment getter.

3. Changes in populateDatabase.xml

- Now, actor HandyWorker1 doesn't have curriculum1 it belongs to HandyWorker2. Curriculum2 belongs to HandyWorker3 and Curriculum3 belongs to HandyWorker4.
- Attachments attribute of complaint5 has been modified in order to test complaints with several attachments.

4. Changes in Repositories

ActorRepository

- Rename method that return a collection of suspicious actors, from isSuspicious to findAllSuspicious.

ApplicationRepository

- ApplicationRepository::findAcceptedApplication has been added.

CreditCardRepository

- Has been deleted.

ComplaintRepository

- Public queries now return Page object instead of Collection object. This helps us to retrieve the complaints in a more efficient way.

- ComplaintRepository::findAllTickers has been deleted.
- ComplaintRepository::findNotSelfAssigned has been renamed to ComplaintRepository::findNotAssigned.
- ComplaintRepository::existTicker has been added.

CurriculumRepository

- Delete query of method existCurriculumByTicker because it's not used nowhere.
- Delete query of method findAllCurriculum because there's an equal method in the repository that returns all the curriculums.
- CurriculumRepository::existTicker has been added.
- CurriculumRepository::findAllTickers has been deleted

FinderRepository

- The method findFixUpTaskFinder(keyWord, startPrice, endPrice, startDate, endDate, warranty, category, pageable) is moved from FinderRepository to FixUpTaskRepository because the method returns a collection of fix-up tasks instead of a Finder or collection of Finders.

FixUpTaskRepository

- findFixUpTaskFinder() query has been moved from FinderRepository to FixUpTaskRepository.
- Public queries now return Page object instead of Collection object. This helps us to retrieve the complaints in a more efficient way.
- FixUpTaskRepository::findAllTickers has been deleted.
- FixUpTaskRepository::existTicker has been added.

PhaseRepository

- Public queries now return Page object instead of Collection object. This helps us to retrieve the complaints in a more efficient way.

SectionRepository

- The methods findHandyWorkerBySection(id) and findTutorialBySection(id) are moved from SectionRepository to HandyWorkerRepository and TutorialSection respectively because both methods don't return Section objets.

5. Changes in Services:

ActorService

- The methods isBanner(actor) and notBanner(actor) has been replaced by changeBanner(actor). ChangeBanner(actor) integrate the functionality of isBanner(actor) and notBanner(actor) in a unique procedure.
- Create and save of RefereeService, HandyWorkerService, AdministratorService, CustomerService and SponsorService has been refactorized into ActorService::createUserAccount and ActorService::save.
- Create method that implement the query of this repository that returns a collection of suspicious actors by the name findAllSuspicious.
- ActorService::definePassword has been removed.
- ActorService::changeBanner has been renamed to ActorService::changeBan

- The following line of ActorService::changeBan has been removed because it is not neccessary: this.userAccountService.save(userAccount);
- ActorService::isSuspicious has been renamed to ActorService::markAsSuspicious
- The following line of ActorService::markAsSuspicious has been removed because it is not neccessary: this.actorRepository.save(actor);

AdministratorService

- AdministratorService::create and AdministratorService::save have been simplified thanks to the refactoring in ActorService.

ApplicationService

- The method checkByPrincipal(application) has been created. It helps to refactoring. It´s invoked twice: ApplicationService::save and ApplicationService::delete.
- The methods removeApplicationToHandyWorker(application) and removeApplicationToFixUpTask(application) have been simplified. Those methods contained unnecessary code.
- ApplicationService::findAcceptedApplication has been added.
- ApplicationService::delete, ApplicationService::removeApplicationToHandyWorker, ApplicationService::removeApplicationToFixUpTask has been deleted due to in the requirements does not specify anything about delete an application.

boxService

- boxService::create has been refactoring so that method is more readable.
- The method checkByPrincipal(box) has been created. It helps to refactoring. It´s invoked twice: boxService::save and boxService::delete.
- The private method has been created: this.checkName(box). It checks that not exists a custom box whose name is “in box”, “out box”, “trash box” and “spam box”. This method is invoked by boxService::save.
- The method createDefaultBox(actor) has been refactorized. Now, it’s more readable.
- The method createDefaultBox(actor) is now a procedure.

CreditCardService

- Has been deleted.

CategoryTranslationService

- The constant named LANGUAGES has been removed. Now, if it’s necessary to know the supported languages by the system, it’s must access to customisation::languages.

ComplaintService

- Public queries now return Page object instead of Collection object. This helps us to retrieve the complaints in a more efficient way.
- ComplaintService::findAllTickers has been deleted.
- ComplaintService::existTicker has been added.
- ComplaintService::findNotSelfAssigned has been renamed to ComplaintService::findNotAssigned.

CurriculumService

- Added the method findByPrincipal(). This method returns the curriculum Principal's. It helps to refactoring.
- Added an assert in findAll method to check the curriculums returned could not be null.
- Delete existCurriculumByTicker method because it's not used nowhere.
- Delete findAllCurriculums method because it's not used nowhere.
- Delete method *delete* because we can't delete a curriculum.
- Deleted findAllTickers method because it's not used nowhere.
- Added existTicker method.

CustomerService

- CustomerService::create and CustomerService::save have been simplified thanks to the refactoring in ActorService.

EducationRecordService

- Added an assert in findAll method to check the education records returned could not be null.
- In EducationRecord::save we take out the line that save the education record in the repository of the if because it's repeated in if condition and else condition.

EndorsementService

- Now, EndorsementService::playedRole is a private method by security reasons.
- The method delete(endorsement) has been update because it's must computing again recipient's score if it has a not null value.

EndorserRecordService

- Added an assert in findAll method to check the curriculums returned could not be null.
- In EndorserRecord::save we take out the line that save the endorser record in the repository of the if because it's repeated in if condition and else condition.

FinderService

- The method checkByPrincipal(finder) has been created. It helps to refactoring. It´s invoked twice: FinderService::save and FinderService::search.

FixUpTaskService

- Added the method checkByPrincipal(fixUpTask): check that the principal can edit or delete the object. It helps to refactoring.
- Added addApplication method. It is used in ApplicationService::save
- findFixUpTaskFinder() method has been move from FinderService to FixUpTaskService.
- Public queries now return Page object instead of Collection object. This helps us to retrieve the complaints in a more efficient way.
- FixUpTaskService::findAllTickers has been deleted.
- FixUpTaskService::existTicker has been added.

HandyWorkerService

- Delete the second parameter in removeCurriculum method because it's not used inside it.
- HandyWorkerService::create and HandyWorkerService::save have been simplified thanks to the refactoring in ActorService.

MessageService

- boxService::create has been refactoring so that method is more readable.
- The method messageToStatus(application, status) has been simplified.

MiscellaneousRecordService

- Added an assert in findAll method to check the curriculums returned could not be null.
- In MiscellaneousRecord::save we take out the line that save the miscellaneous record in the repository of the if because it's repeated in if condition and else condition.
- In MiscellaneousRecord::delete we delete an assert that checks the miscellaneous record we want to delete exist in the repository because there's another one to check the id of this miscellaneous record is not equal to 0.

PersonalRecordService

- Added an assert in findAll method to check the curriculums returned could not be null.
- In PersonalRecord::save we take out the line that save the personal record in the repository of the if because it's repeated in if condition and else condition.

PhaseService

- Public queries now return Page object instead of Collection object. This helps us to retrieve the complaints in a more efficient way.

ProfessionalRecordService

- Added an assert in findAll method to check the professional records returned could not be null.
- In ProfessionalRecord::save we take out the line that save the professional record in the repository of the if because it's repeated in if condition and else condition.
- In ProfessionalRecord::delete we delete an assert that checks the professional record we want to delete exist in the repository because there's another one to check the id of this professional record is not equal to 0.

RefereeService

- RefereeService::create and RefereeService::save have been simplified thanks to the refactoring in ActorService.
- RefereeService::principalHasSelfAssigned has been added. We will use this method in the controllers of Complaints.

SectionService

- Added the method checkByPrincipal(section): check that the principal can edit or delete the object. It helps to refactoring.

SocialProfileService

- Added the method checkByPrincipal(socialProfile): check that the principal can edit or delete the object. It helps to refactoring.

SponsorService

- SponsorService::create and SponsorService::save have been simplified thanks to the refactoring in ActorService.

SponsorshipService

- Added the method checkByPrincipal(sponsorship): check that the principal can edit or delete the object. It helps to refactoring.

TutorialService

- Added the method checkByPrincipal(tutorial): check that the principal can edit or delete the object. It helps to refactoring.

PersonalRecordService, EducationallRecordService, ProfessionalRecordService, MiscellaneousRecordService and EndorserRecordService

- Added checkByPrincipal(TRecord): this procedure check if the principal can edit or delete the record "TRecord" received as argument.
- The methods save() and deleted() invoke checkByPrincipal().

UtilityService

- The method checkDate(start, end) has been created: check that start moment must be before than end moment. This methos is invoked in different services several times: EducationRecordService::save, ProfessionalRecordService::save, PhaseService::save, FixUpTaskService::save, NoteService::save and ReportService::save.
- The method current_moment() has been created. It helps to refactoring. It is invoked in several services: ApplicationService,::save TutorialService::save, FinderService::save, FixUpTaskService::save, ComplaintService::save, ReportService::save, NoteService::save, EndorsementService::save and MessageService::save.
- The procedure checkName(actor) has been added. It checks if name actor's is not "System". CustomerService::save, AdministratorService::save, SponsorService::save, HandyWorkerService::save and RefereeService::save use it.
- The method UtilityService::generateValidTicker now also check if the new ticker generated is already in use in any FixUpTask or Complaint.
- UtilityService::getSplittedAttachments has been added. This will be useful in several controllers.