

1 Problem statement

We need to extend our solution so that the hiring and firing of mechanics are now managed, in addition to their payrolls. All use cases referring to the user manager and user interfaces, as well as de options for all of them will be provided.

1.1 Use cases and system actors

Manager	<ul style="list-style-type: none"> • Extended mechanic management • Contract management • Payroll management • Contract type management • Profesional groups management
---------	--

Each of the new use cases is detailed below.

2 Extended mechanic management

The base use case is modified according to the following:

- Carry out **verifications in the remove operation of each mechanic**. The previously existing remove operation of each mechanic (see reference statement) is slightly affected. A mechanic can only be deleted if he has no work orders (in any state), interventions, or contracts.
- A new menu option will be added (**List mechanics with a valid contract**) that will show all the mechanics with a valid contract.

3 Contract management

3.1 Add a contract to a mechanic

There is an option of the menu to create a contract for a mechanic.

The manager will enter the mechanic's **DNI**, the type of contract (**contractTypeName**), the identifier of the professional group (**professionalGroupName**) and the annual base wage (**annualBaseWage**). All fields are required.

In the case that the contract has a fixed duration (**FIXED_TERM**), the manager must also enter a contract end date which, in addition, must be later than the start date. Additionally, the salary must be greater than 0.

The mechanic, the contract type and the professional group **must exist in the system**. If it does not exist, the user will be notified.

In addition to all these fields, each contract will be in one of the following **states**: **IN_FORCE** (the initial state when creating a contract) or **TERMINATED**.

There is also a date (month and year) of **contract starting** (**startDate**) which will be set when a contract is created, and a **date (month and day) when the contract ends** (**endDate**), which will be set when it ends.

We will assume that the duration of the contracts is multiples of a month. Therefore, when registering a contract, the **contract start date will be set to the first day of the next month** and the contract end date will be the last day of the month in which the contract ends.

Although for fixed-term contracts (FIXED_TERM) the manager must enter an end date of the contract, this is merely indicative and the effective date of the end of the contract will be the one in which the necessary conditions are met to end the contract (see section ??).

The application must keep all the contracts that the mechanics have had over the time, even though for a given mechanic there can only be one in force.

A special case occurs when, when registering a contract for a mechanic, **a contract already exists for this mechanic**. In this case, the following considerations will be followed:

- The current contract becomes finished on the last day of the current month.
- The corresponding compensation will be calculated for the termination of the contract.
- The new contract becomes the effective contract dated the first day of the following month.

3.2 Delete a contract

A contract may be deleted as long as, during its validity, the mechanic has not carried out any activity (*workorders*) nor has he/she had payrolls (*payrolls*) generated from that contract.

There will be a menu option that shows all existing contracts and asks the manager which one to delete, providing the contract id.

3.3 Update a contract

Once a contract is registered, while it is in force, errors made during the registration process could be corrected. Therefore, it is allowed to modify the expected end date and the annual base salary if the contract is of fixed duration (FIXED_TERM). The effects of the change will be applicable from the moment of the modification.

There is a menu option that shows all existing contracts and asks the manager which one to delete, providing the contract id.

3.4 Contract listings

The manager will be able to consult at any time all the contracts, all those of a worker or a specific contract.

- In case of selecting the option to show a certain contract, all the existing contracts are shown to the manager and he is asked which one to show providing the contract id. All contract details are displayed.
- In case of showing all the contracts of a worker, the manager must enter the worker's ID. For each one, a selection of the contract fields will be shown: the id, the mechanic's identity card, the status, how many payrolls have been received and the amount of the settlement if it is in the finished state.
- If the option to show all the contracts is selected, a summary of the fields of the registered contracts is also displayed for each one, as in the previous point.

3.5 Finish a contract

A contract will always end for one of the following two reasons:

1. a new one is created for the mechanic (see section ??), or
2. the manager executes the menu option to end a contract.

In both cases, the end date of the contract will be established as the last day of the current month and the settlement amount of the current contract, if any, will be calculated. The menu option provided shows all the contracts and asks the manager to enter one of the id's. The contract will be terminated.

3.6 Compensation for termination of contract

When a contract ends, the mechanic must be **compensated at the end of the contract**. A mechanic will be entitled to this compensation if you have worked for at least **one full year**.

This compensation at the end of the contract will be a payment (not a salary) to which no discount will be applied and will be saved together with the rest of the contract data (**settlement**).

To calculate it, you need to know

- A) the average daily gross salary received in the last 12 months. It is obtained by adding the gross totals (without applying discounts) of the last 12 payrolls and dividing by 365 days,
- B) the number of compensation days to which you are entitled per year worked, and
- C) the number of full years elapsed since the beginning of the contract.

Knowing all this, the **compensation will be calculated as $A*B*C$**

4 Payroll management

4.1 Payroll generation

There is an option in the manager menu to generate the payroll for the current month.

A mechanic will receive 14 payments per year. One for each month, plus two extraordinary ones in the months of June and December. The 14 payments will be received in 12 payrolls.

To implement the payroll generation process, the following must be taken into account:

- Each month the payroll is generated for those employees who either **have a valid contract**, or **their contract expires this month**,
- **as long as there is no payroll already generated that month**. In that case, **no new payroll will be generated**.
- Each payroll will include concepts of **earnings** and **deductions**. The sum of the payroll payments is called **gross wage**, the sum of the deductions is called **deductions** and the final amount to be charged is called **net wage**.
- **Payment**

- **Base salary, monthlyWage.** The result of dividing the annual base salary by the 14 payments.
- **Extra pay, bonus.** In the corresponding months, with an amount equal to the base salary for that month.
- **Productivity Plus, productivityBonus.** It is a salary supplement that is calculated as a percentage of the total amount of work orders assigned to the mechanic that have been opened in this payroll month and that have already been invoiced. This percentage depends on your professional group.
- **Triennials, trienniumPayment.** Salary supplement for seniority in the company that the mechanics will receive monthly. For every three full years accumulated in the same contract in force, a three-year period will be accumulated. For each three years, the mechanic will receive an amount that depends on his professional group.

- **Discounts**

- For IRPF tax (**incomeTax**). The annual income is divided by tranches and each one corresponds to a percentage. The current sections are as follows:
 - * From 0€ to 12.450€: 19%.
 - * From 12.450€ to 20.2000€: 24%.
 - * From 20.200€ to 35.200€: 30%.
 - * From 35.200€ to 60.000€: 37%.
 - * From 60.000 to 300.000€: 45%.
 - * More than 300.000€: 47%.

The IRPF tax deduction will be calculated by applying the corresponding percentage (**rate**) to the total gross payroll.

- By contribution to Social Security (National Insurance Contribution, **NIC**). It will be 5% of the annual base salary prorated over 12 months.

When generating the payroll, **each of the credit and discount concepts** indicated above must be specified.

4.2 Delete payrolls

Two possible functions will be implemented to delete payroll:

- delete a mechanic's payroll for the current month, if any, and
- delete all payroll generated in the current month, if any exist.

In the first case, the manager is asked to enter the mechanic's ID.

4.3 Payroll listings

The manager will be able to consult the payroll according to several criteria:

1. **List all payrolls.** It shows a summary of each one with its identifier, version, date (month and year) and the net total received.
2. **Detail of a specific payroll.** It must show all the fields of the payroll, including the date (month and year), gross total, breakdown by concept of the total gross, total of discounts, breakdown by concept of the total discounts and total net.

3. **List of payroll received by an employee.** It shows the same summary as in point one.
4. **List of payroll received by the mechanics of a professional group.** It shows the same summary as in point one.

5 Profesional groups management

The collective agreement negotiated in the company will establish the **professional groups**, which may vary over time, as new agreements are negotiated.

Each professional group is defined by a **name** and must be unique, the **amount to be charged for each triennium (triennium payment)** and the **productivity bonus percentage**. The professional group **must necessarily appear in the contract** since it influences the salary by determining the amount per three years and the productivity bonus.

It is necessary to implement a CRUD of professional groups in the following way:

- When creating a new professional group, the user must provide the **professional group name (name)**, the amount of the **trienniums (trienniumSalary)** and the percentage of the productivity bonus (**productivityRate**). All fields are required. The string field cannot be null or empty and numeric fields must be greater than or equal to zero.
- It will not be possible to remove a professional group that has associated contracts, whether or not they are in force.
- All the data of the professional group can be modified, except the identifier and the name. The changes made will be taken into account in future payrolls and will not affect those already generated.
- Workers with a valid contract may be listed in a professional group. At the end of the list, the total number of workers belonging to this professional group will be shown, as well as an accumulated figure with the total annual base salary of these workers.

Initially, there are the following professional groups:

```
(id, version, name, trienniumSalary, productivityRate)
('I',1,'I',46.74E0,5.0E0)
('II',1,'II',38.12E0,4.5E0)
('III',1,'III',33.44E0,3.0E0)
('IV',1,'IV',28.85E0,3.5E0)
('V',1,'V',19.64E0,2.5E0)
('VI',1,'VI',14.78E0,2.0E0)
('VII',1,'VII',11.25E0,1.5E0)
```

6 Contract management

Although the types of contract may vary over time, depending on the legislation, there will always be, at a minimum, contracts **PERMANENT**, **FIXED_TERM** and **SEASONAL**. However, the application manager may register new types of contract to adjust to the current legislation.

The type of contract is identified by a unique name and will determine the **number of compensation days per year worked** to which the worker is entitled in the event of termination of the contract.

A CRUD of contract types will be implemented with the following characteristics:

- When adding a contract type, it will be asked the **name of the contract type** and the **number of compensation days (compensationDays)** at the end of the contract. All fields are required. The string field cannot be null or empty and numeric fields must be greater than or equal to zero.
- Workers with a valid contract of one type may be listed. At the end of the list, the total number of workers with this type of contract will be shown, as well as an accumulated figure with the total annual base salary of these workers.
- Contract types that have associated contracts cannot be deleted.
- All contract type data except your id and name can be changed and the changes will be effective in the future.

There are some default values:

```
(id, version, name, compensationDays)
('FIXED_TERM',1,'FIXED_TERM',4.2)
('PERMANENT',1,'PERMANENT',1.35)
('SEASONAL',1,'SEASONAL',3.25)
```

7 To consider

A test project is provided that allows you to validate many aspects of your implementation. For the tests to be successful, it is necessary that all the decimal values that are going to be stored in a field of any Dto of the business layer, be truncated to two decimal places.

8 Use cases you must implement

- Exercise number 0 if (your UO % 3 == 0):
 - Payroll management
 - Contract type management
 - Extended mechanic management
- Si tu UO módulo 3 es 1 (uo % 3 == 1):
 - Contract management
 - Contract type management
- Si tu UO módulo 3 es 2 (uo % 3 == 2):
 - Payroll management
 - Profesional groups management
 - Extended mechanic management

Regardless of the use cases that touch you, it is necessary that the domain model be implemented completely and correctly for all of them (as if you had to implement them all). To make up for the lack of a use case that maintains the corresponding data, the necessary records will be inserted in the database.

9 Correction criteria

A series of **mandatory** criteria are mentioned below. Not using them will have a **penalty** in the grade.

- The design of the application should conform to the **package structure** seen in class.
- All submitted code will go through a **plagiarism detector**.
- You need to do the **handling of errors** (exceptions) correctly and do it in the right place (you can't interact with the user anywhere other than the presentation layer). It is not valid for the application to terminate abruptly with an exception trace on the screen.
- All operations must keep the **referential integrity of the data**. Beware of deletions or modifications (ex: what happens if I try to delete a mechanic with contract(s)? You should not allow it).
- The code must conform to the **Java Code Conventions**¹. It must not have any warnings (they will be treated as compilation errors).
- The **code** must be *clean*, well-indented, legible, and properly commented (javadoc where applicable). There should be no *dead* code, no debugging code (*println*, ...).
- The application should work properly **at least with the HSQLDB engine**. JUnit tests are provided. **Make sure they all pass green BEFORE submitting**.
- All SQL/JPQL statements must be **externalized**.
- A single compressed file must be delivered with the following content:
 - Complete **Eclipse project** (without the bin)
 - Within the Eclipse project, you must include a text file whose name will be your UO code (U0nnnnn.txt) with the **personal information** (Name, DNI), the **use cases** that should be implemented and **comments** you want to make.
 - The util project must not be delivered, nor is it allowed to modify it. If you consider that any method of this library can be improved, or you miss one, do it in a new package within the project that is delivered. If you think it is worth adding it to the original library, you can comment it in the U0nnnnn.txt file.

10 PLEASE, TAKE IN MIND

There is a forum in Campus Virtual for questions about the course, this project, possible solutions, etc., where you can ask questions, as well as the teacher in the classroom and office hours. Use all these resources as much as you need. The forum is a very useful tool so that teachers can realize about possible problems that arise and thus be able to help you. This project, as many others, will take you some time. The teachers have estimated, and verified after several years, that the time to be devoted fits the program and the teaching guide. But keep in mind that these estimation assumes you are doing it at the pace planned, in parallel with the classroom sessions of theory and laboratory, and that you dedicate the expected autonomous working time each week (1,5 hours for each classroom hour). Leaving all work for the last week completely invalidates that planning and the teaching methodology we are trying to implement; even more, the most probable result, will be that you will waste a lot of time trying to get the point of things, you will find infinite minor problems that you do not know how to solve (of which the teacher almost certainly warned you in the classroom,

¹<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

but you do not longer remember...), make a mess reading through forums, the anxiety to deliver leads you to copy fragments of code that you find out there, but that you do not fully understand, and that, in the end, you are stuck and do not understand the conceptual essence of the subject. After the submission, you must do an authorship proof of the project you submitted and the theory test. The proof is intended to prove that you implemented the project while the test usually consists of short questions about concepts that you had to face in practice. Even if you could submit a project, it will be of no help if you did not get the basic concepts. Our advice is keeping the pace of the classroom sessions and the schedule the teachers suggest.