# Requirement’s documentation

## Requirements List

### Functional requirements

The Back Office Operator must be able to create new payment orders, specifying a bill by a list of candidate payment orders.

The Back Office Operator can issue not-issued payment orders.  
When a payment order is issued, it is given a protocol number generated incrementally.

The Back Office Operator can delete not-issued payment orders.

The Back Office Operator can reissue suspended payment orders.  
When a payment order is reissued his protocol number is not changed.

The Back Office Operator can save as not pertinent suspended payment orders.

The Back Office Operator can save as paid notified payment orders.

The Back Office Operator can save as suspended notified payment orders.

When a payment order is issued the system must generate and store a PDF document with all the useful communications for the contributor.

A user must be able to log in as a Back Office Operator in desktop application.

A user must be able to log in as a Readings Operator in mobile application.

The Readings Operator must have the list of the assigned readings updated automatically.

The Readings Operator can save readings specifying a meter and the water consumed.

The Readings Operator must be able to send saved readings via Internet.

### Non-functional Requirements

Payment orders can be seen and searched specifying a protocol, a debtor, a year, a trimester and/or a status.

The readings are first saved locally on smartphone and sent later only if internet connection is available.

To use the system every user must log in.

### Domain Requirements

Not issued payment orders can only be issued or deleted.

Issued payment orders can only be archived.

Once saved as paid or saved as not pertinent, a payment order must be archivied.

Suspended payment orders can only be reissued or saved as not pertinent.

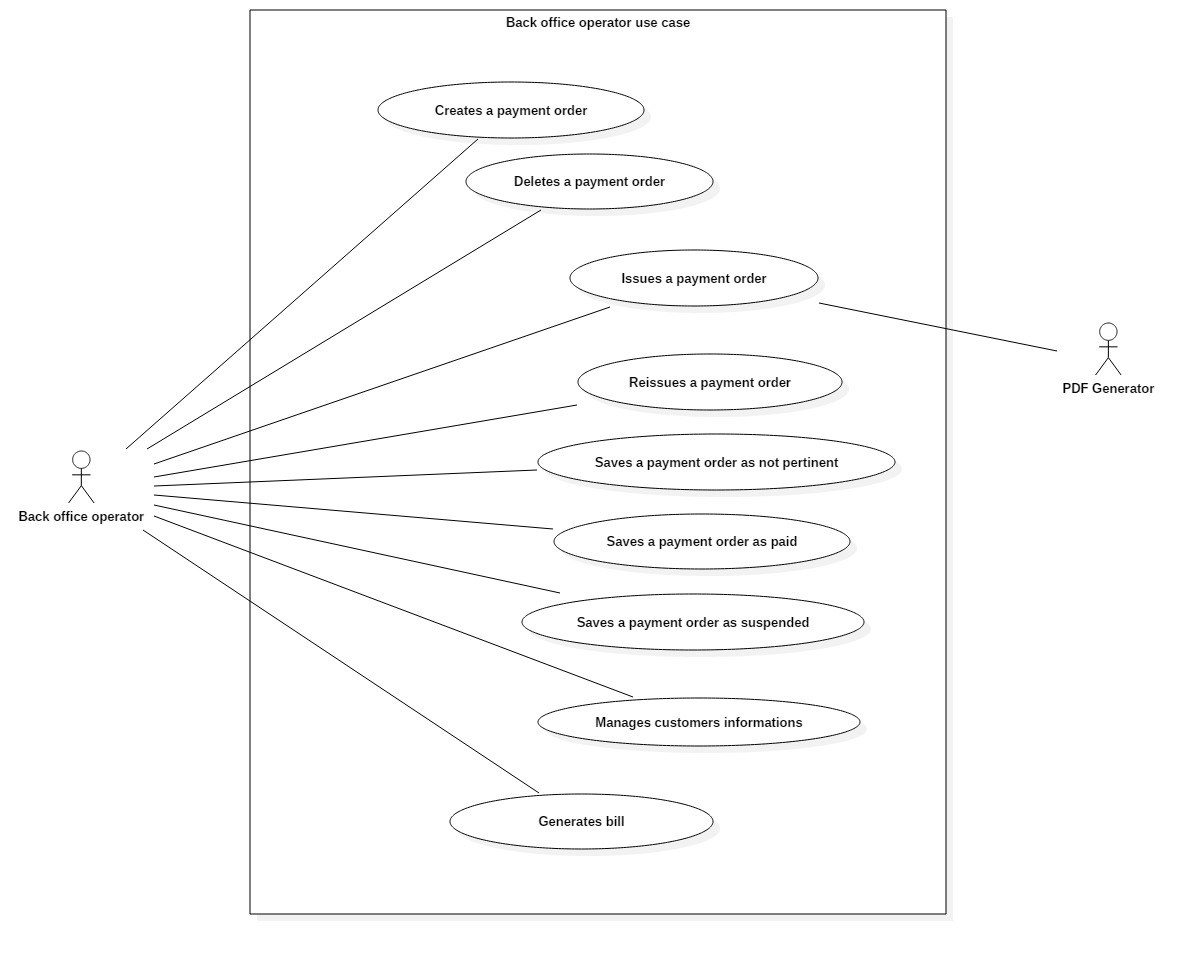
Every issued payment order is identified by a protocol number.

Notified payment orders can only be saved as paid or as suspended.

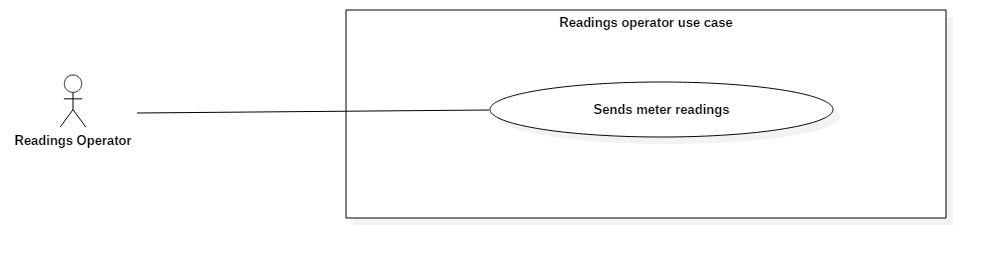
The considered unit of measurement of water is m3.

## Use Case Diagrams

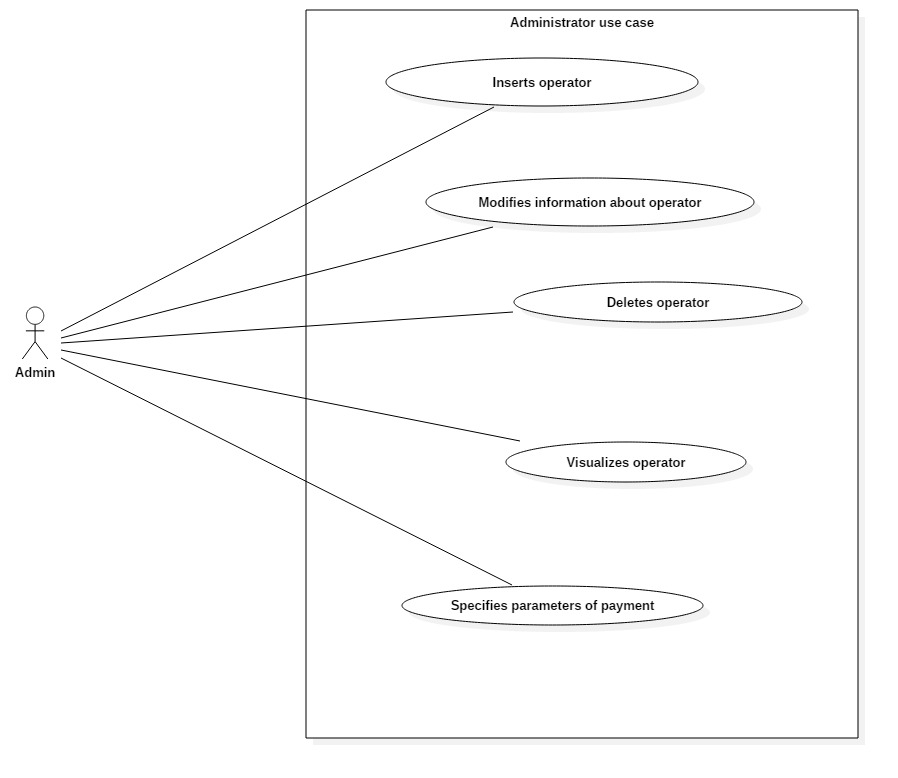
### Back office operator use case



### Readings operator use case



### Administrator use case



## Cockburn Diagrams for Use Cases

### Create payment order

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE *#1* | *Create payment order* | | |
| Goal in Context | Generate a new payment order. | | |
| Scope & Level | System under design ; Level = User goal | | |
| Preconditions | The user must be logged in the system. | | |
| Success End Condition | A payment order is created and stored in the system. | | |
| Failed End Condition | The user presses “Home” in “Payment Orders” mockup. | | |
| Primary Actor | Back Office Operator. | | |
| Trigger | The operator presses “Payment Orders” in “Main Menu” mockup. | | |
| DESCRIPTION | **Step n°** | **Back Office Operator** | **System** |
| *1* |  | Shows mockup “Payment Orders” |
| *2* | Presses “New payment order” |  |
| *3* |  | Shows mockup “Candidate Payment Orders” |
| *4* | Selects a bill |  |
| *5* |  | Enables “Create Payment Order” |
| *6* | Presses “Create Payment Order” |  |
| *7* |  | Shows mockup “Ask Create” |
| *8* | Presses “Yes” |  |
| *9* |  | Shows mockup “Operation Success” |
| *10* | Presses “OK” |  |
|  | *11* |  | *Returns to mockup “Payment Orders”* |

|  |  |  |  |
| --- | --- | --- | --- |
| EXTENSIONS | **Step n°** | **Back Office Operator** | **System** |
|  | *2.1* | Presses “Home” |  |
|  | *2.2* |  | *Shows mockup “Main Menu”* |
| SUBVARIATIONS | *Step* | **Back Office Operator** | **System** |
|  | *2.2* | Sets filtering parameters and presses “Filter” |  |
|  | *3.2* |  | *Shows the filtered table and return to step #2* |

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE *#2* | *Delete payment order* | | |
| Goal in Context | Delete a selected payment order. | | |
| Scope & Level | System under design ; Level = User goal | | |
| Preconditions | The user must be logged in the system. | | |
| Success End Condition | A payment order is deleted from the system. | | |
| Failed End Condition | The user presses “Home” in “Payment Orders” mockup. | | |
| Primary Actor | Back Office Operator. | | |
| Trigger | The operator presses “Payment Orders” in “Main Menu” mockup. | | |
| DESCRIPTION | **Step n°** | **Back Office Operator** | **System** |
| *1* |  | Shows mockup “Payment Orders” |
| *2* | Selects a payment order |  |
| *3* |  | Enables “Delete” |
| *4* | Presses “Delete” |  |
| *5* |  | Shows mockup “Ask Delete” |
| *6* | Presses “Yes” |  |

### Delete payment order

|  |  |  |  |
| --- | --- | --- | --- |
|  | *7* |  | Shows mockup “Operation Success” |
| *10* | Presses “OK” |  |
|  | *11* |  | *Returns to mockup “Payment Orders”* |
| EXTENSIONS | **Step n°** | **Back Office Operator** | **System** |
|  | *2.1* | Presses “Home” |  |
|  | *3.1* |  | *Shows mockup “Main Menu”* |
| SUBVARIATIONS | *Step* | **Back Office Operator** | **System** |
|  | *2.2* | Sets filtering parameters and presses “Filter” |  |
|  | *3.2* |  | *Shows the filtered table and return to step #2* |

Aggiornamenti:

Cambiare numeri di step estensioni (FATTO)

Da mandare (FATTO)

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE *#5* | Archives payment orders. | | |
| Goal in Context | User archives a suspended payment order. | | |
| Scope & Level | System Under Design; Level = “User goal” | | |
| Preconditions | User must be logged. | | |
| Success End Condition | A payment order is archived. | | |
| Failed End Condition | User presses ‘Home’ in mockup ‘Payment Orders’. | | |
| Primary Actor | Back office operator. | | |
| Trigger | Back office operator presses ‘Payment Orders’ in mockup ‘Main menu’. | | |
| DESCRIPTION | **Step n°** | **Actor 1** | **System** |
| *1* | Presses ‘Payment Orders’ in mockup ‘Main menu’. |  |
| *2* |  | Shows mockup ‘Payment Orders’. |
| *3* | Selects a suspended payment order. |  |
| *4* |  | Enables button ‘Archive’. |
| *5* | Presses ‘Archive’. |  |
| *6* |  | Shows mockup ‘Confirm operation’. |
| *7* | Presses ‘Yes’. |  |
| *8* |  | Shows mockup ‘Successful operation’. |
| *9* | Presses ‘Ok’. |  |
| *10* |  | Shows mockup ‘Payment Orders’ in which there isn’t the payment order concerned because it shall be archived, and UC successfully ends. |
| EXTENSIONS | ***Step*** | **Actor 1** | **System** |
|  | *\*.1* | Presses ‘Home’ in mockup ‘Payment Orders’. |  |
| *\*2.1* |  | Shows mockup ‘Main menu’ and UC fails. |
|  | *5.2* | Presses ‘No. |  |
| *6.2* |  | Back to step 2. |
| SUBVARIATIONS | ***Step*** | **Actor 1** | **System** |
|  | *3.3* | Filters suspended payment order, selects it. |  |
| *4.3* |  | Back to step 4. |

## User Interface Mockups

## Entity-Boundary-Control Class Diagram

## Sequence Diagrams

## State Chart

## Glossary (A-Z)

**Back Office Operator** User of desktop application, his goal is to handle the entire payment order management system.

**Issue** Render a payment order effective. After issuing a payment order, it must be paid by the debtor.

**Not pertinent** If the debtor won the legal case, the payment order would be archived. The debtor won’t have to pay the payment order anymore.

**Payment order** Document which requires the debtor to pay within a defined deadline.

**Protocol number** Every issued payment order is labeled by an index, a protocol number, which is considered a unique identifier.

**Suspend** A payment order can be suspended in presence of a legal case.

**Reading** Reading of water consumption reported on the meter.

**Reading Operator** User of mobile application.