



GCI'16 Documentation

Manfredi Vincenzo N86001560

Pollastro Andrea N86001233

Santangelo Andrea N86001703

Scalella Andrea Francesco N86001625



This page is intentionally left blank.

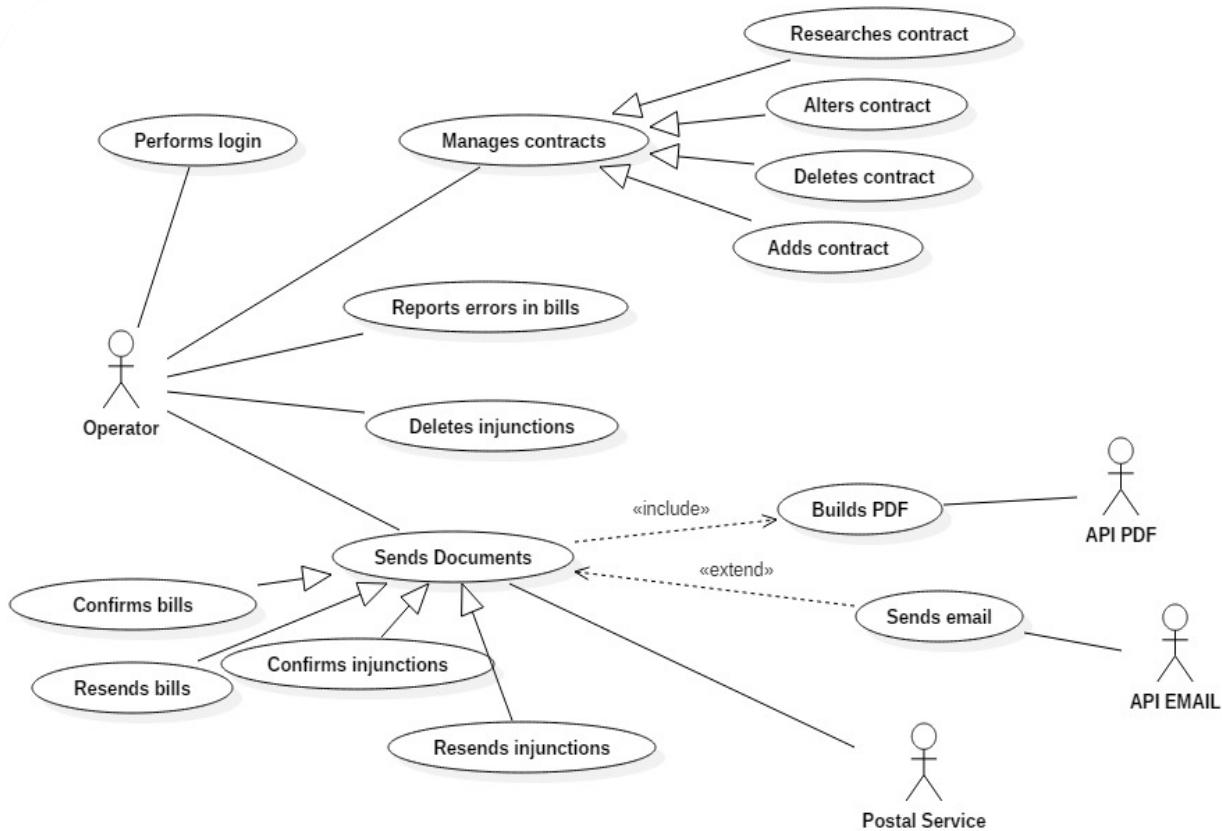
Contents

| | |
|--|-----|
| Software requirements document | 2 |
| <i>Functional model</i> | 2 |
| Use case diagram..... | 2 |
| Cockburn tables..... | 3 |
| Mockup | 14 |
| Gantt diagram..... | 43 |
| <i>Domain model</i> | 44 |
| Analysis class diagram | 50 |
| <i>CRC Cards</i> | 44 |
| <i>Analysis sequence diagrams.....</i> | 51 |
| System design document | 60 |
| <i>Architecture analysis.....</i> | 60 |
| <i>Technologies</i> | 61 |
| <i>Design pattern & Implementation choices</i> | 62 |
| <i>Design sequence diagram</i> | 66 |
| <i>Design class diagram</i> | 95 |
| Testing document | 104 |
| <i>System testing</i> | 104 |
| <i>Performs login test</i> | 104 |
| <i>Researches contact test</i> | 105 |
| <i>Alters contract test</i> | 106 |
| <i>Deletes contract test</i> | 107 |
| <i>Deletes contract test</i> | 108 |
| <i>Adds contract test</i> | 109 |
| <i>Reports errors in bills test</i> | 110 |
| <i>Deletes injunctions test.....</i> | 111 |
| <i>Confirms bill test</i> | 112 |
| <i>Resends bill test</i> | 113 |
| <i>Confirms injunctions test.....</i> | 114 |
| <i>Resends injunctions test.....</i> | 115 |
| <i>JUnit code</i> | 116 |
| <i>Source.....</i> | 116 |
| <i>Results</i> | 120 |

Software requirements document

Functional model

Use case diagram



Cockburn tables

Performs login

| | |
|------------------------------|---|
| Use case | Performs login |
| Goal in context | The operator wants to authenticate |
| Preconditions | The goal is to let the operator log into the system inserting the right credentials |
| Success and condition | The operator gets access to the system |
| Failed end condition | The operator can't access to the system |
| Operator | Operator |
| Trigger | The operator starts the program |

MAIN SCENARIO

| Step n. | Operator | System |
|----------------|---|---|
| 1 | The operator starts the program | |
| 2 | | The system shows the "Login" mockup |
| 3 | The operator fills all field in the "Login" mockup | |
| 4 | | The system enables the "Login" button in the "Login" mockup |
| 5 | The operator presses the "Login" button in the "Login" mockup | |
| 6 | | The system shows the "Home" mockup |

EXTENSION n.1

| Step n. | Operator | System |
|----------------|--|---|
| 6.1 | | The system shows the "Login – error" mockup |
| 7.1 | The operator presses the "ok" button in the "Login – error" mockup | |
| 8.1 | | Go to step 3 |

Researches contract

| | |
|------------------------------|---|
| Use case | Researches contract |
| Goal in context | The operator can be able to read information about the contract researched |
| Preconditions | The operator must be logged in and he has compiled almost one field in the “Registry management” mockup |
| Success and condition | The operator reads the informations about the contract researched |
| Failed end condition | There aren’t contracts stored into the system |
| Operator | Operator |
| Trigger | The operator clicks on “Search” button in the “Registry management” mockup |

MAIN SCENARIO

| Step n. | Operator | System |
|----------------|--|--|
| 1 | The operator clicks on “Search” button in the “Registry management” mockup | |
| 2 | | The system shows the details of the contracts founded into the table |

EXTENSION #1

| Step n. | Operator | System |
|----------------|--|---|
| 2.1 | | The system shows the “Registry management – error” mockup |
| 3.1 | The operator clicks on the “Ok” button in the “Registry management – error” mockup | |

SUBVARIATION #1

| Step n. | Operator | System |
|----------------|--------------------------------------|--|
| 1.2 | The operator leaves all fields blank | |
| 2.2 | | The system shows details of all contracts stored into the system |

Alters contract

| | |
|------------------------------|---|
| Use case | Alter contract |
| Goal in context | Modify a pre-existing contract |
| Preconditions | The operator must be logged in, he has searched a contract and he has selected it from the table in the “Registry management” mockup |
| Success and condition | The operator has modified a pre-existing contract |
| Failed end condition | The operator fills the interested fields with invalid characters. The operator cancels the operation and system shows the “Registry management” mockup |
| Operator | Operator |
| Trigger | The operator clicks on the “Alter holder” button in the “Registry management” mockup |

MAIN SCENARIO

| Step n. | Operator | System |
|----------------|--|--|
| 1 | The operator clicks on “Alter holder” button in the “Registry management” mockup | |
| 2 | | The system shows the “Alter holder” mockup |
| 3 | The operator edits one or more fields | |
| 4 | | The system enables the “Alter” button associated with the field edited |
| 5 | The operator clicks on the “Alter” button | |
| 6 | | The system shows the “Alter holder – success” mockup |
| 7 | The operator clicks on the “Ok” button in the “Alter holder – success” mockup | |

EXTENSION #1

| Step n. | Operator | System |
|----------------|---|---|
| 6.1 | | The system shows the “Alter holder – error” mockup with a reference to the form that contains the error |
| 7.1 | The operator clicks on the “Ok” button in the “Alter holder – error” mockup | |
| 8.1 | | Go to step 2 |

Deletes contract

| | |
|------------------------------|--|
| Use case | Deletes contract |
| Goal in context | The operator wants to delete a contract |
| Preconditions | The operator must be logged in and he has selected a contract from the table contained in the mockup “Registry management” |
| Success and condition | The operator deletes the selected contract |
| Failed end condition | The operator can't delete the selected contract The operator cancels the operation |
| Operator | Operator |
| Trigger | The operator clicks on the “Delete” button in the “Registry management” mockup |

MAIN SCENARIO

| Step n. | Operator | System |
|----------------|--|---|
| 1 | The operator clicks on “Delete” button in the “Registry management” mockup | |
| 2 | | The system shows the “Delete contract” mockup |
| 3 | The operator clicks on the “Yes” button in the “Delete contract” mockup | |
| 4 | | The system shows the “Delete contract – success” mockup |
| 5 | The operator clicks on the “Ok” button in the “Delete contract – success” mockup | |
| 6 | | The system shows the “Registry management” mockup |

EXTENSION #1

| Step n. | Operator | System |
|----------------|--|---|
| 3.1 | | The system shows the “Delete contract – error” mockup |
| 4.1 | The operator clicks on the “Ok” button in the “Delete contract – error” mockup | |
| 5.1 | | Go to step 6 |

EXTENSION #2

| Step n. | Operator | System |
|----------------|--|---------------|
| 3.2 | The operator clicks on the “No” button in the “Delete contract” mockup | |
| 4.2 | | Go to step 6 |

Report errors in bills

| | |
|------------------------------|---|
| Use case | Report errors in bills |
| Goal in context | The operator wants to report errors in bills |
| Preconditions | The operator must be logged in and he has selected a bill in the bills' table of the "Bills queue" mockup |
| Success and condition | The operator reports an error in a bill |
| Failed end condition | The operator clicks on the "Cancel" button |
| Operator | Operator |
| Trigger | The operator clicks on the "Report error" button in the "Bills" mockup |

MAIN SCENARIO

| Step n. | Operator | System |
|---------|---|---|
| 1 | The operator clicks on "Report error" button in the "Bills" mockup | |
| 2 | | The system shows the "Report error" mockup |
| 3 | The operator fills the "Report's specifications" field in the "Report error" mockup | |
| 4 | | The system enables the "Send" button in the "Report error" mockup |
| 5 | The operator clicks on the "Send" button in the "Report error" mockup | |
| 6 | | The system shows the "Send" mockup |

EXTENSION #1

| Step n. | Operator | System |
|---------|---|--------------|
| 3.1 | The operator clicks on the "Cancel" button in the "Report error" mockup | |
| 4.1 | | Go to step 8 |

EXTENSION #2

| Step n. | Operator | System |
|---------|---|--------------|
| 5.2 | The operator clicks on the "Cancel" button in the "Report error" mockup | |
| 6.2 | | Go to step 8 |

Deletes injunction

| | |
|------------------------------|--|
| Use case | Deletes injunction |
| Goal in context | The operator wants to delete an injunction that has the “non-issued” state |
| Preconditions | The operator must be logged in and he has selected an injunction from the injunctions’ table in the “Injunctions queue” mockup |
| Success and condition | The operator deletes one or more injunctions from the injunctions’ table in the “Injunctions queue” mockup |
| Failed end condition | The operator doesn’t delete the injunctions from the table |
| Operator | Operator |
| Trigger | The operator clicks on the “Delete” button in the “Injunctions queue” mockup |

MAIN SCENARIO

| Step n. | Operator | System |
|---------|--|---|
| 1 | The operator clicks on the “Delete” button in the “Injunctions queue” mockup | |
| 2 | | The system shows the “Delete injunction – success” mockup |
| 3 | The operator clicks on the “Ok” button | |
| 4 | | The system shows the “Injunctions queue” mockup |

EXTENSION #1

| Step n. | Operator | System |
|---------|--|---|
| 2.1 | | The system shows the “Delete injunction – error” mockup |
| 3.1 | The operator clicks on the “Ok” button in the “Delete injunction – error” mockup | |

Confirms bills

| | |
|------------------------------|---|
| Use case | Confirms bills |
| Goal in context | The operator wants to confirm a single or a group of bills |
| Preconditions | The operator must be logged in and he has selected a bill (or more than one bill) in the bills' table of the "Bills queue" mockup |
| Success and condition | The operator confirms a bill |
| Failed end condition | The operator cancels the operation |
| Operator | Operator |
| Trigger | The operator clicks on the "Confirm" button in the "Bills queue" mockup |

MAIN SCENARIO

| Step n. | Operator | System |
|----------------|---|---|
| 1 | The operator clicks on the "Confirm" button in the "Bills queue" mockup | |
| 2 | | The system shows the "Build PDF" mockup |
| 3 | The operator clicks on the "Send PDF" button in the "Build PDF" mockup | |
| 4 | | The system shows the "Send PDF" mockup |
| 5 | The operator clicks on the "Ok" button | |
| 6 | | The system shows the "Bills queue" mockup |

EXTENSION #1

| Step n. | Operator | System |
|----------------|--|---------------|
| 3.1 | The operator clicks on the "Cancel" button in the "Build PDF" mockup | |
| 3.1 | | Go to step 6 |

EXTENSION #2

| Step n. | Operator | System |
|----------------|-----------------|---|
| 4.2 | | The system shows the "Send PDF" mockup in which the field "Log address' error" is visible |
| 5.2 | Go to step 5 | |

SUBVARIATION #1

| Step n. | Operator | System |
|----------------|---|--|
| 1.3 | The operator clicks on the "Confirm" button in the "Bills queue" mockup after he has selected more than one row from the table or he has clicked on the "Select all" button | |
| 2.3 | | The system shows the "Build PDF – multiple" mockup |
| 3.3 | The operator clicks on the "Send PDFs" button in the "Build PDF – multiple" mockup | |
| 4.3 | | The system shows the "Send PDF – multiple" mockup |
| 5.3 | Go to step 5 | |

Resends bills

| | |
|------------------------------|---|
| Use case | Resends bills |
| Goal in context | The operator wants to resend a bill that has the “Issued” state |
| Preconditions | The operator must be logged in and he has selected a bill in the bills’ table of the “Bills” mockup |
| Success and condition | The operator must be logged in and he has selected a bill in the bills’ table of the “Bills” mockup |
| Failed end condition | The operator cancels the operation The PDF interface is not available |
| Operator | Operator |
| Trigger | The operator clicks on the “Build PDF” button in the “Bills” mockup |

MAIN SCENARIO

| Step n. | Operator | System |
|----------------|--|---|
| 1 | The operator clicks on the “Build PDF” button in the “Bills” mockup | |
| 2 | | The system shows the “Build PDF” mockup |
| 3 | The operator clicks on the “Send PDF” button in the “Build PDF” mockup | |
| 4 | | The system shows the “Send PDF” mockup |
| 5 | The operator clicks on the “Ok” button | |
| 6 | | The system shows the “Registry management” mockup |

EXTENSION #1

| Step n. | Operator | System |
|----------------|-----------------|--|
| 4.1 | | The system shows the “Send PDF” mockup in which the “Log address’ error” form is visible |
| 5.1 | Go to step 5 | |

EXTENSION #2

| Step n. | Operator | System |
|----------------|--|-------------------------------------|
| 3.2 | The operator clicks on the “Cancel” button in the “Build PDF” mockup | |
| 4.2 | | The system shows the “Bills” mockup |

EXTENSION #2

| Step n. | Operator | System |
|----------------|--|---|
| 2.3 | | The system shows the “Build PDF – error” mockup |
| 3.3 | The operator clicks on the “Ok” button in the “Build PDF – error” mockup | |
| 4.3 | | Go to step 6 |

Confirms injunctions

| | |
|------------------------------|---|
| Use case | Confirms injunctions |
| Goal in context | The operator wants to send an injunction to the customer |
| Preconditions | The operator must be logged in and he has selected an injunction from the table contained in “Injunctions queue” mockup |
| Success and condition | The system creates and sends the PDF |
| Failed end condition | The operator cancels the operation The PDF interface is not available |
| Operator | Operator |
| Trigger | The operator clicks on the “Confirm” button in the “Injunction queue” mockup |

MAIN SCENARIO

| Step n. | Operator | System |
|----------------|--|---|
| 1 | The operator clicks on the “Confirm” button in the “Injunction queue” mockup | |
| 2 | | The system shows the “Build PDF” mockup |
| 3 | The operator clicks on the “Send PDF” button in the “Build PDF” mockup | |
| 4 | | The system shows the “Send PDF” mockup |
| 5 | The operator clicks on the “Ok” button in the “Send PDF” mockup | |
| 6 | | The system shows the “Registry management” mockup |

EXTENSION #1

| Step n. | Operator | System |
|----------------|--|---|
| 3.1 | The operator clicks on the “Cancel” button in the “Build PDF” mockup | |
| 4.1 | | The system shows the “Injunctions queue” mockup |

EXTENSION #2

| Step n. | Operator | System |
|----------------|-----------------|---|
| 4.2 | | The system shows the “Send PDF” mockup in which the “Log address” error form is visible |
| 5.2 | Go to step 5 | |

EXTENSION #3

| Step n. | Operator | System |
|----------------|--|---|
| 2.3 | | The system shows the “Build PDF – error” mockup |
| 3.3 | The operator clicks on the “Ok” button in the “Build PDF – error” mockup | |
| 4.3 | | Go to step 6 |

Resends injunctions

| | |
|------------------------------|---|
| Use case | Resends injunctions |
| Goal in context | The operator wants to resend an injunction that has the “Issued” state |
| Preconditions | The operator must be logged in and he has selected an injunction in the injunctions’ table of the “Injunction” mockup |
| Success and condition | The operator resends an injunction that has the “Issued” state |
| Failed end condition | The operator cancels the operation |
| Operator | Operator |
| Trigger | The operator clicks on the “Build PDF” button in the “Injunction” mockup |

MAIN SCENARIO

| Step n. | Operator | System |
|----------------|--|---|
| 1 | The operator clicks on “Build PDF” button in the “Injunction” mockup | |
| 2 | | The system shows the “Build PDF” mockup |
| 3 | The operator clicks on the “Send PDF” button in the “Build PDF” mockup | |
| 4 | | The system shows the “Send PDF” mockup |
| 5 | The operator clicks on the “Ok” button in the “Send PDF” mockup | |
| 6 | | The system shows the “Registry management” mockup |

EXTENSION #1

| Step n. | Operator | System |
|----------------|--|--|
| 3.1 | The operator clicks on the “Cancel” button in the “Build PDF” mockup | |
| 4.1 | | The system shows the “Injunction” mockup |

EXTENSION #2

| Step n. | Operator | System |
|----------------|-----------------|---|
| 4.2 | | The system shows the “Send PDF” mockup in which the “Log address” error form is visible |
| 5.2 | Go to step 5 | |

EXTENSION #3

| Step n. | Operator | System |
|----------------|--|---|
| 2.3 | | The system shows the “Build PDF - error” mockup |
| 3.3 | The operator clicks on the “Ok” button in the “Build PDF – error” mockup | |
| 4.3 | | Go to step 6 |

Adds contract

| | |
|------------------------------|---|
| Use case | Adds contract |
| Goal in context | Create a new contract |
| Preconditions | The operator must be logged in and he clicked the “Registry management” button from the “Home” mockup |
| Success and condition | The operator adds a new contract |
| Failed end condition | The operator doesn’t fill all fields The operator cancels the operation |
| Operator | Operator |
| Trigger | The operator clicks on the “Add” button in the “Registry management” mockup |

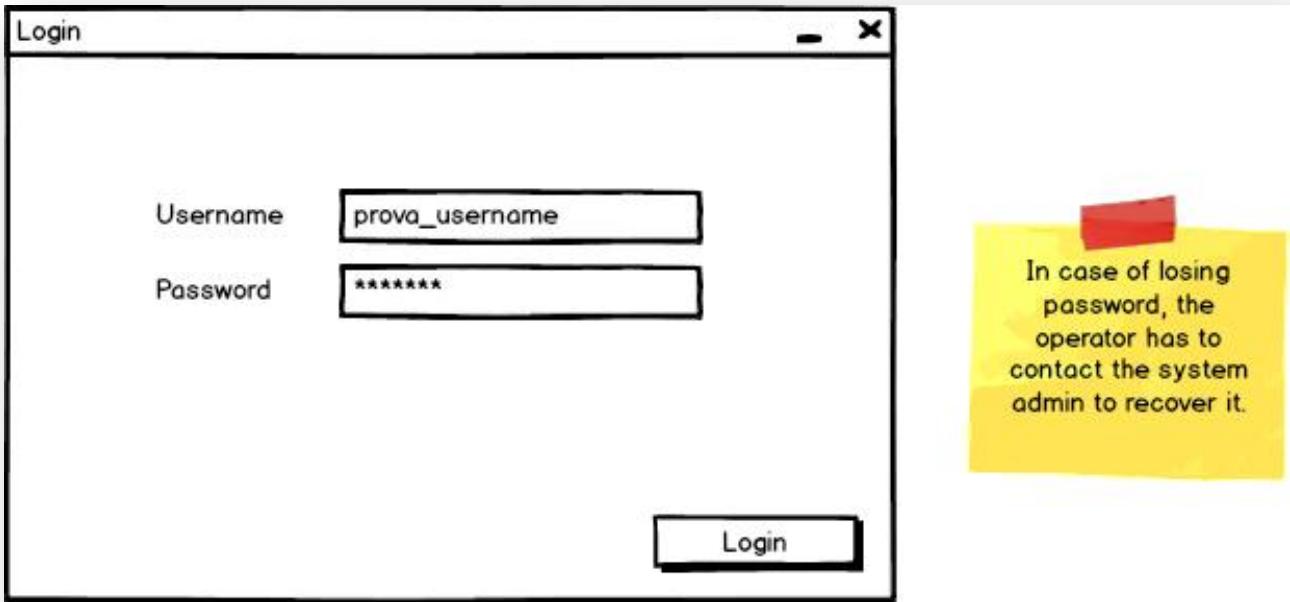
MAIN SCENARIO

| Step n. | Operator | System |
|----------------|--|--|
| 1 | The operator clicks on “Add” button in the “Registry management” mockup | |
| 2 | | The system shows the “Add holder” mockup |
| 3 | The operator fills all fields and clicks the “Add” button in the “Add holder” mockup | |
| 4 | | The system shows the “Add holder – success” mockup |
| 5 | The operator clicks on the “Ok” button in the “Add holder – success” mockup | |
| 6 | | The system shows the “Add holder” mockup |

EXTENSION #1

| Step n. | Operator | System |
|----------------|---|--|
| 4.1 | | The system shows the “Add holder – error” mockup with reference to the field that contains error |
| 5.1 | The operator clicks on the “Ok” button in the “Add holder – error” mockup | |
| 6.1 | | Go to step 2 |

Mockup



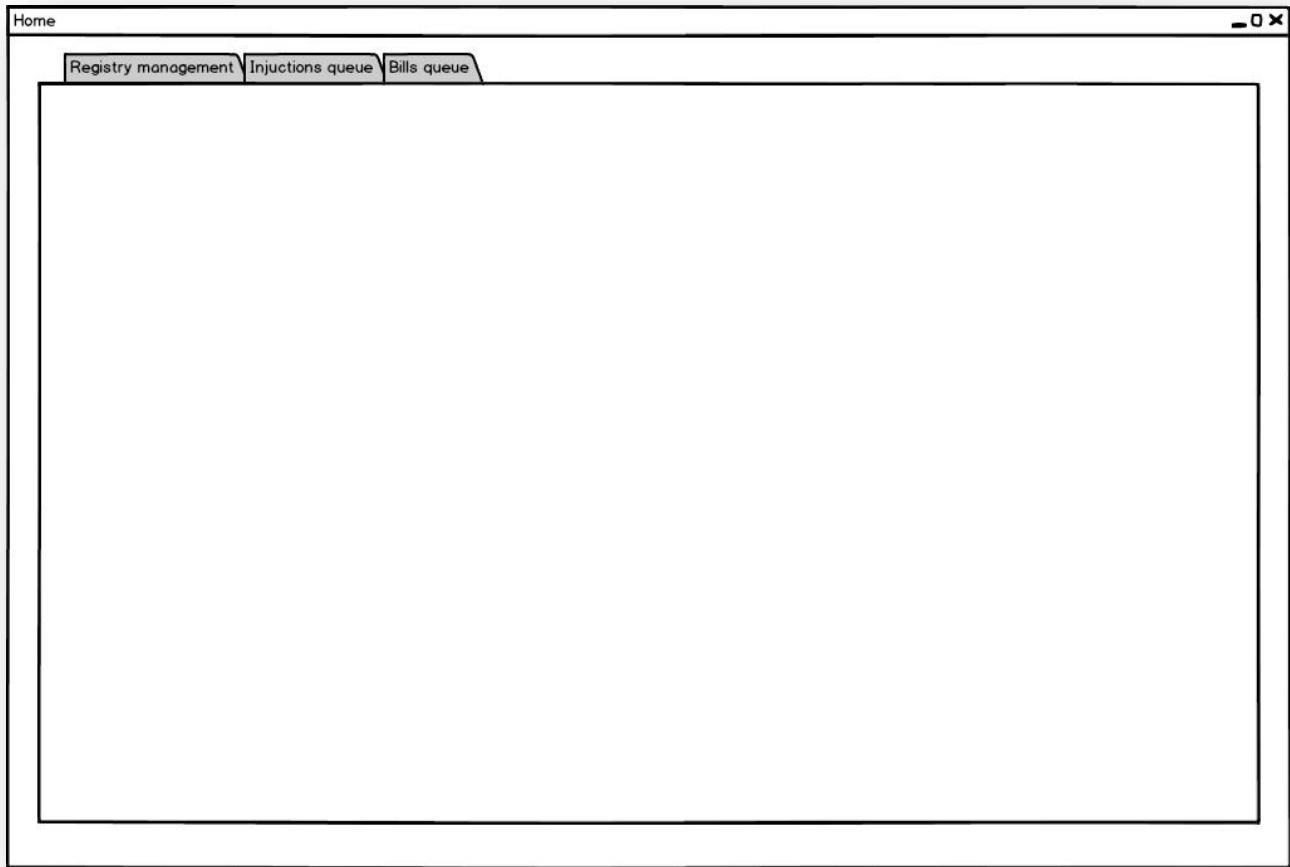
A screenshot of a Windows-style login window titled "Login". It contains two text input fields: "Username" with the value "prova_username" and "Password" with the value "*****". Below the inputs is a "Login" button. To the right of the window is a yellow sticky note with a red pushpin at the top. The note contains the following text:

In case of losing password, the operator has to contact the system admin to recover it.

Login



Login - error



Home

To make a new search it's enough to fill one or more field and click on "Search" button.

| Name | Surname | Contract ID | Tax C./VAT |
|---------|------------|-------------|------------|
| Giacomo | Guilizzoni | 12345 | mng24f |
| Marco | Botton | 34526 | fge45 |
| Mariah | MacLachian | 65381 | hgr35 |
| Valerie | Liberty | 94562 | fre79 |

Add

Search

Alter holder

Remove

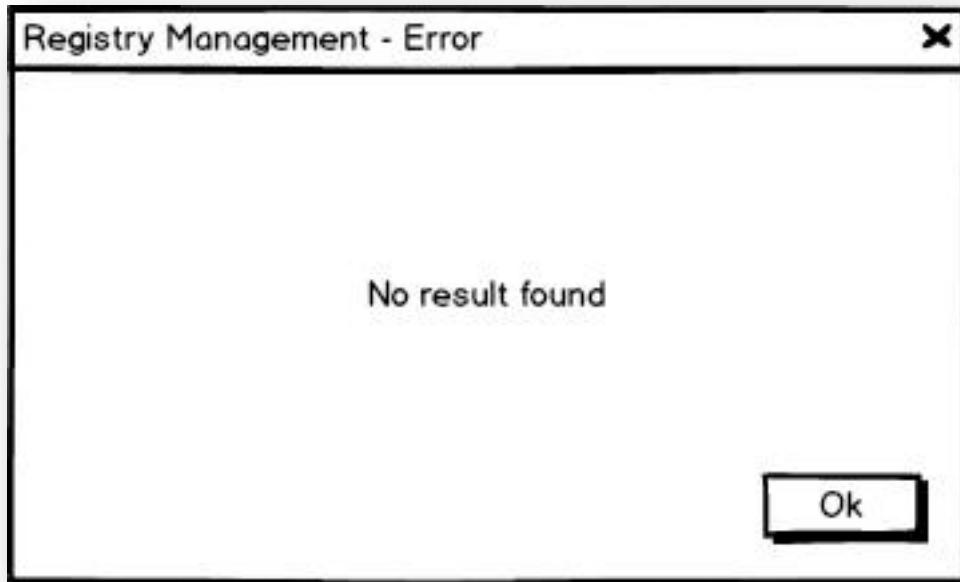
Billing address: Via esempio n. 10, 80127 Napoli, NA
Address: Via esempio2 n. 12, 80127 Napoli, NA
Telephone: 0000000000
eMail: esempio@example.it

Bills

Injunctions

After have selected a single row from the table the grey area will be usable.

Registry management



Registry management – error

After gave selected a single row from the table the gray area will be usable.

Injunctions queue

Bills queue - Single

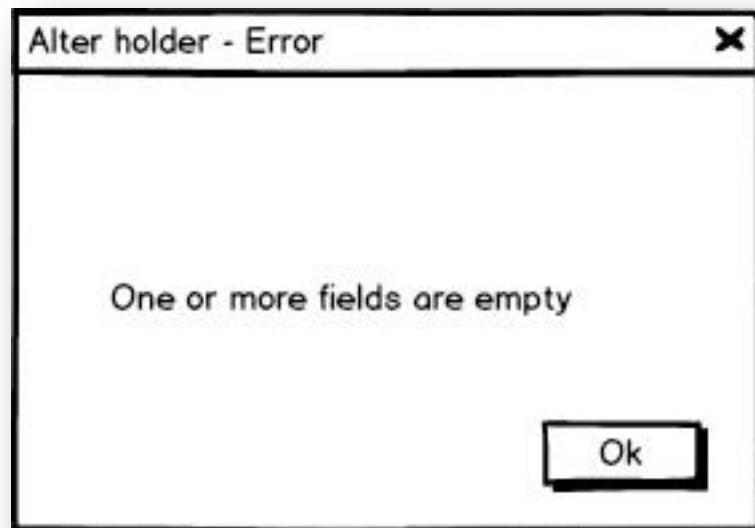
Bills queue - Multiple

Alter holder

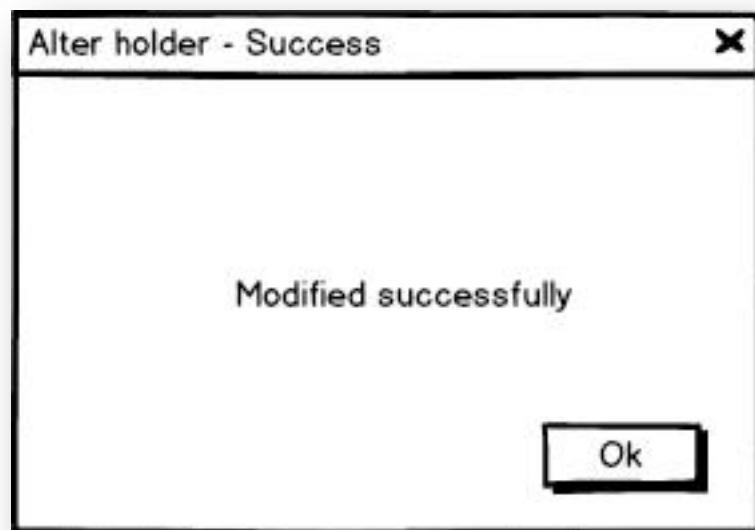
| | |
|--|---|
| Anagrafica | Bill address |
| Name <input type="text" value="Mario"/> | Street <input type="text" value="Via Cintia"/> |
| Surname <input type="text" value="Rossi"/> | Number <input type="text" value="21"/> |
| Tax C./VAT <input type="text" value="RSRSMR020A65F839P"/> | District <input type="text" value="NA"/> |
| Telephone <input type="text" value="0000000000"/> | Postal code <input type="text" value="80100"/> |
| eMail <input type="text" value="esempio@example.it"/> | |
| | Address |
| | Street <input type="text" value="Via Roma"/> |
| | Number <input type="text" value="40"/> |
| | District <input type="text" value="NA"/> |
| | Postal code <input type="text" value="80100"/> |
| <input type="button" value="Indietro"/> | <input type="button" value="Alter"/> |
| <input type="button" value="Alter"/> | <input type="button" value="Alter"/> |

Each "Alter" button is associated to the respective fields

Alter holder



Alter holder - error

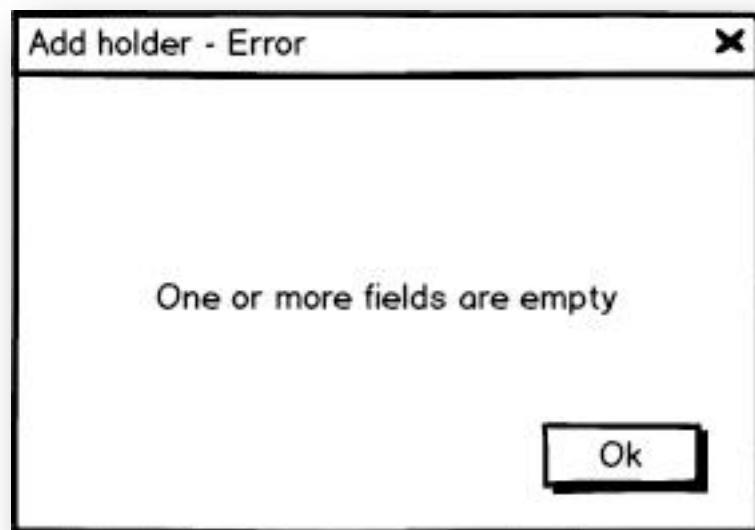


Alter holder - success

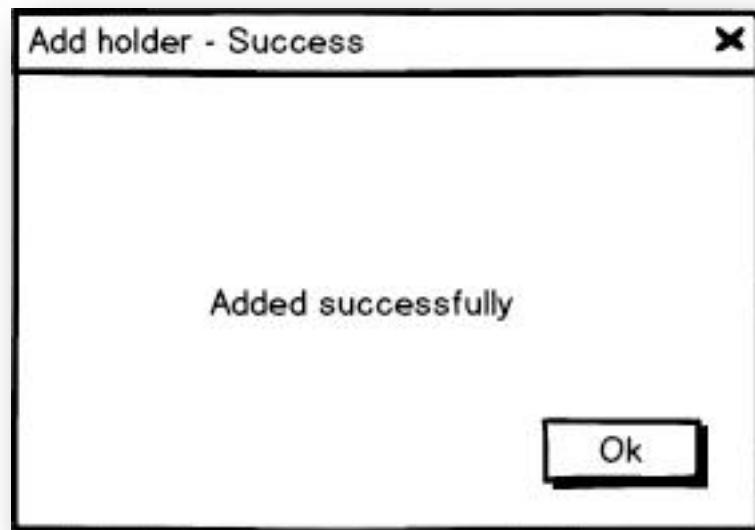
Add holder

| | |
|-------------------------------------|---------------------------------------|
| Registry | Bills address |
| Name <input type="text"/> | Street <input type="text"/> |
| Surname <input type="text"/> | Number <input type="text"/> |
| Tax C./VAT <input type="text"/> | Postal code <input type="text"/> |
| Telephone <input type="text"/> | City <input type="text"/> |
| eMail <input type="text"/> | District <input type="text"/> |
| Address | |
| Street <input type="text"/> | Number <input type="text"/> |
| City <input type="text"/> | Postal code <input type="text"/> |
| <input type="button" value="Back"/> | <input type="button" value="Create"/> |

Add holder



Add holder - error



Add holder - success

Bills

| Invoice N. | Period | State | Due date |
|------------|---------------------|--------|----------|
| 51985 | 02/01/01 - 01/03/01 | Paid | 16/03/01 |
| 94872 | 02/03/01 - 01/06/01 | Issued | 16/06/01 |
| | | | |
| | | | |

Consumption data

Tax (€ x m³): 0,05

Total (€): 5,00

Detection (m³): 100

Operator ID: 12345

Detection date: 01/01/2016

Report error

Build PDF

Back

After have selected a single row from the table, the grey area will be usable.

Bills

Injunctions

| Injection number | Bill ID | State |
|------------------|---------|--------|
| 1 | 51985 | Issued |
| 2 | 94872 | Payed |
| | | |
| | | |

Summary

Bill's period of reference : 01/01/2016 - 01/03/2016
Bill due date: 15/03/2016
Detection date: 16/03/2016
Payment date: 20/03/2016

[Back](#) [Build PDF](#)

After have selected a single row from the table the grey area will be usable.

Injunctions

Report error

Summary

Rate (€ x m³): 0,05
Total (€): 6,00
Detection (m³): 100
Operator ID: 12345
Detection date: 01/01/2016
Due date: 15/03/2016

Report's specifications

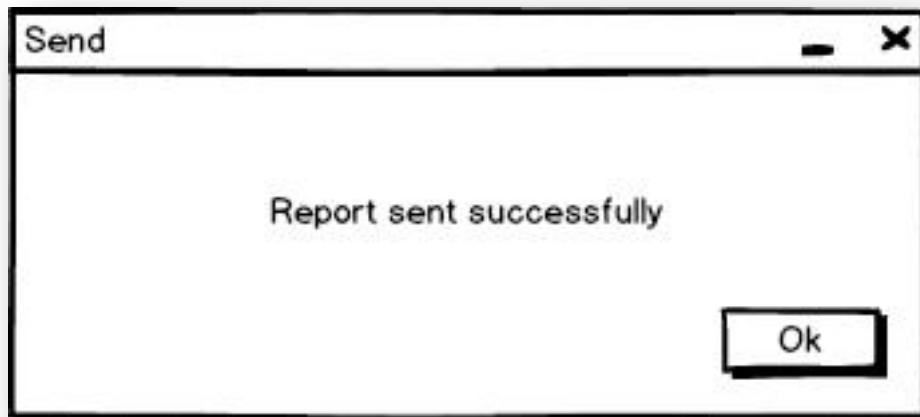
The total value is wrong

Cancel Send

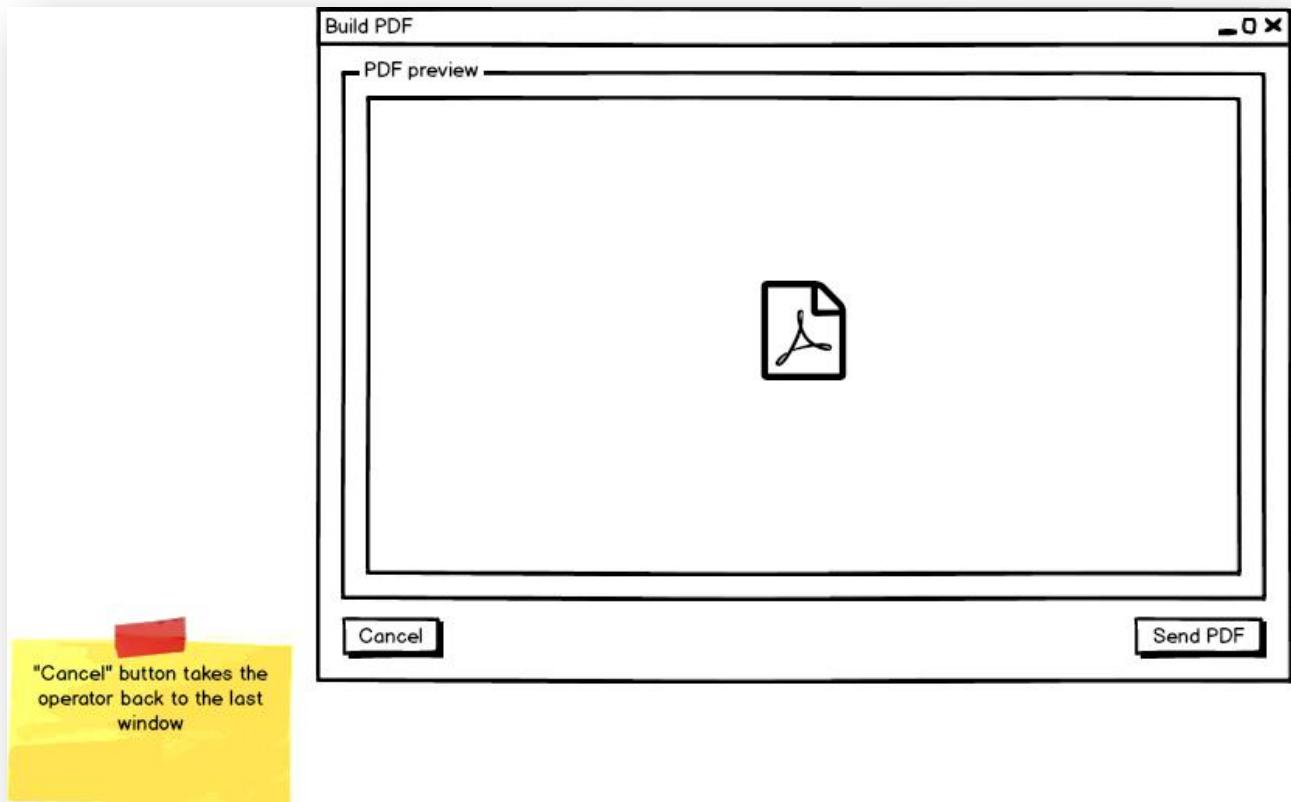
The "Report's specifications" field must be filled

After filled the "Report's specifications" field, the grey area will be usable.

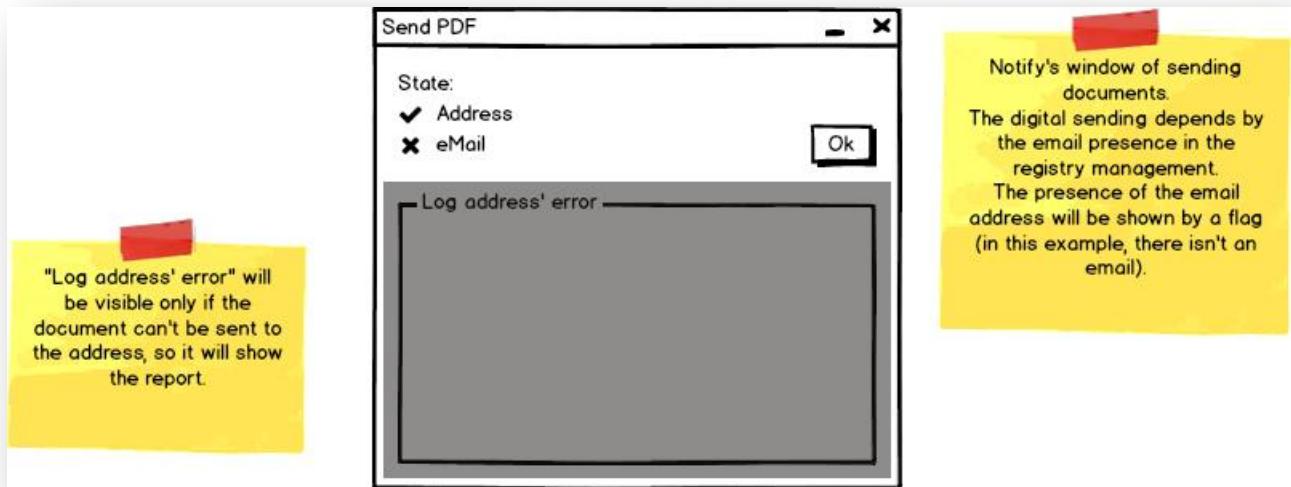
Report error



Send



Build PDF



Send PDF

Build PDF - Multiple

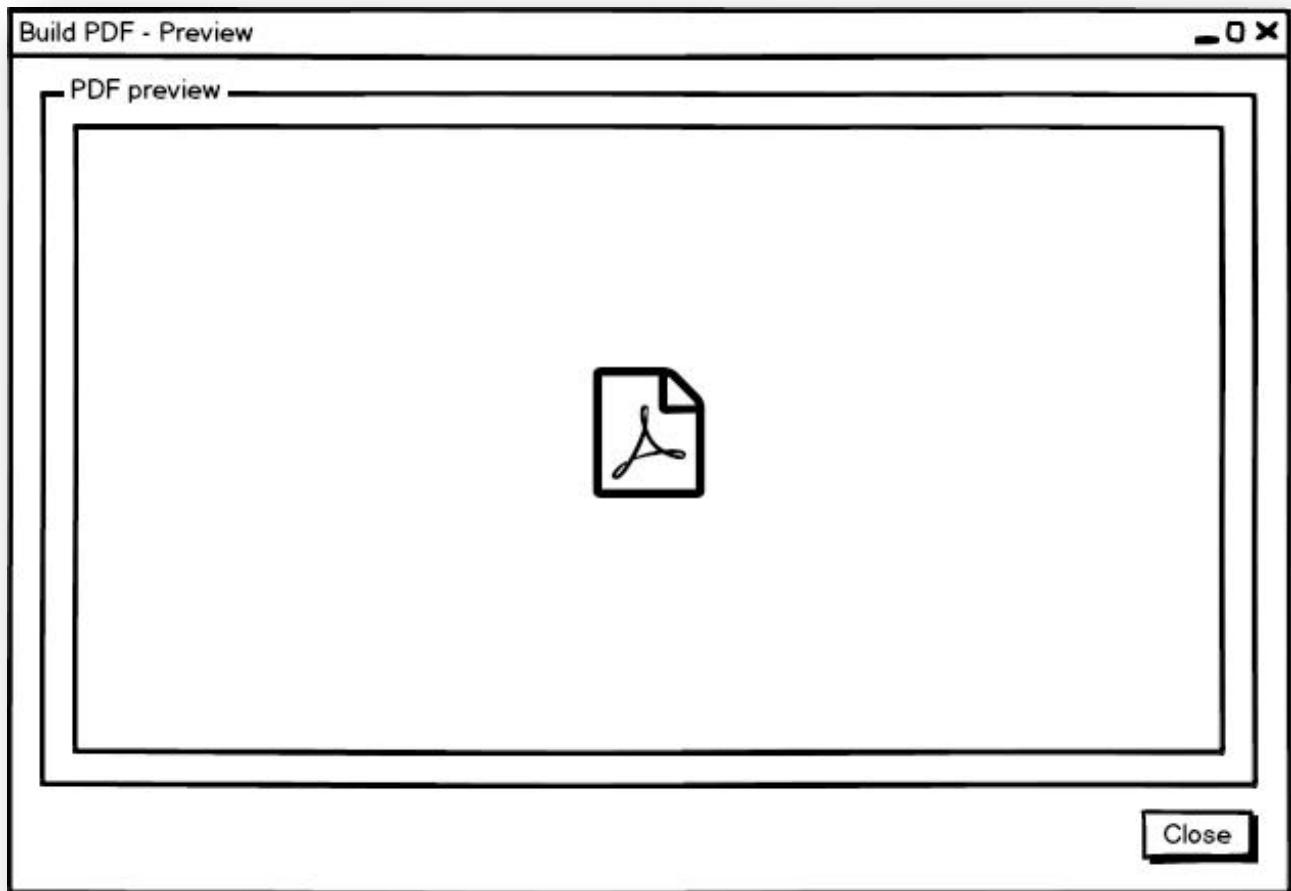
| Contract ID | Reference detection | Generated on | Total |
|-------------|---------------------|--------------|--------|
| 12345 | 1 | 03/01/2016 | € 9.00 |
| 54321 | 2 | 04/01/2016 | € 3.00 |
| | | | |
| | | | |
| | | | |
| | | | |

Preview PDF

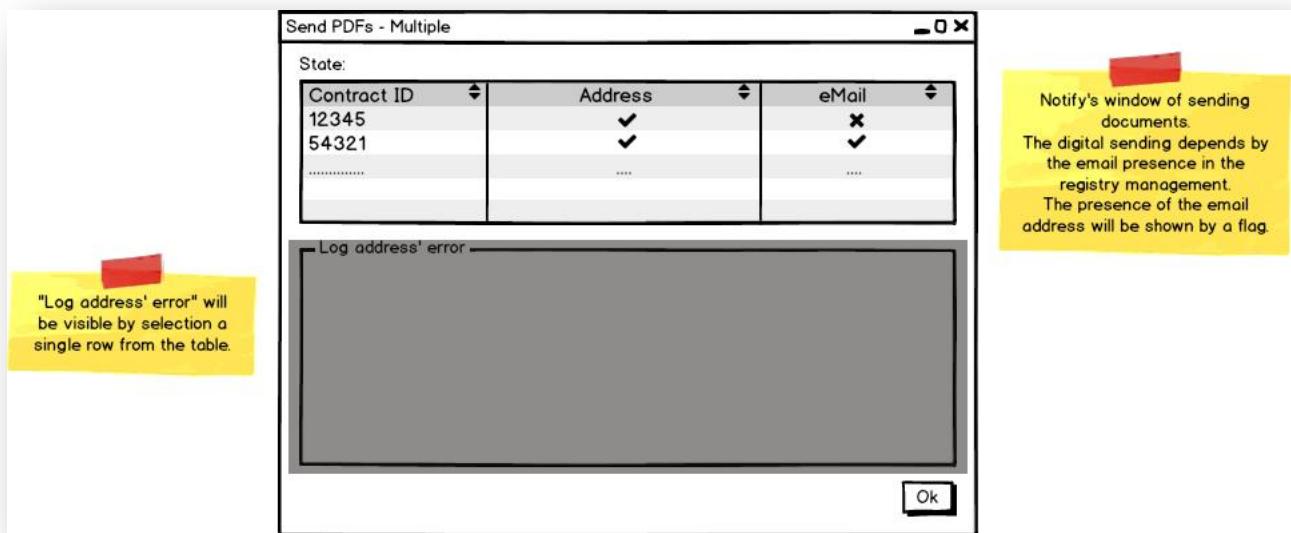
Back Send PDFs

After have selected a single row from the table the grey area will be usable.

Build PDF - multiple



Build PDF - preview



Send PDFs - multiple

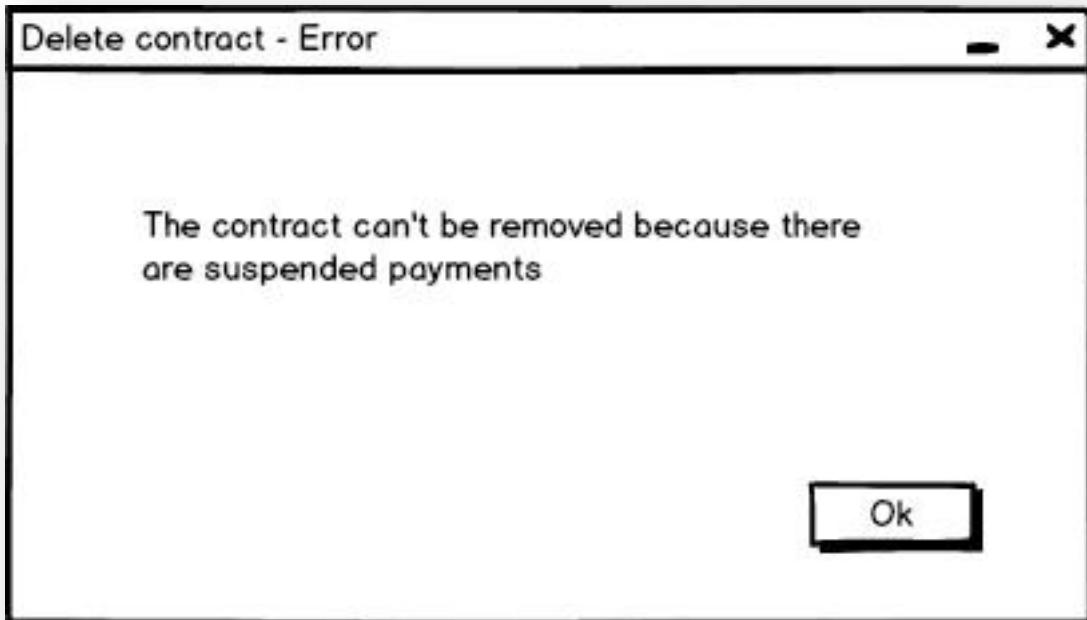
Alert

Are you sure you want to delete the contract?

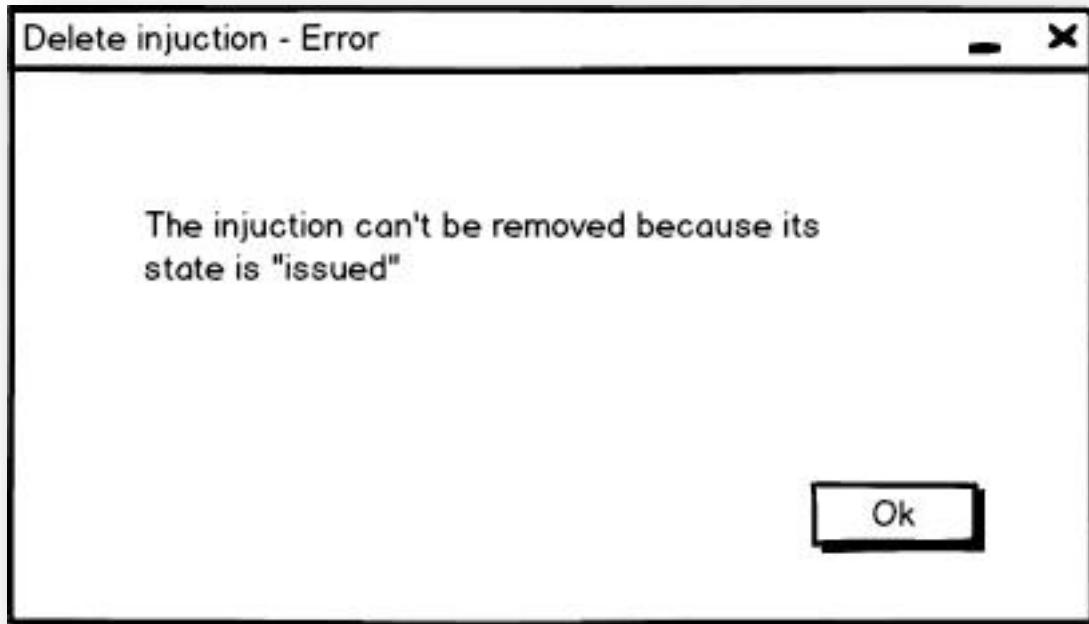
No

Yes

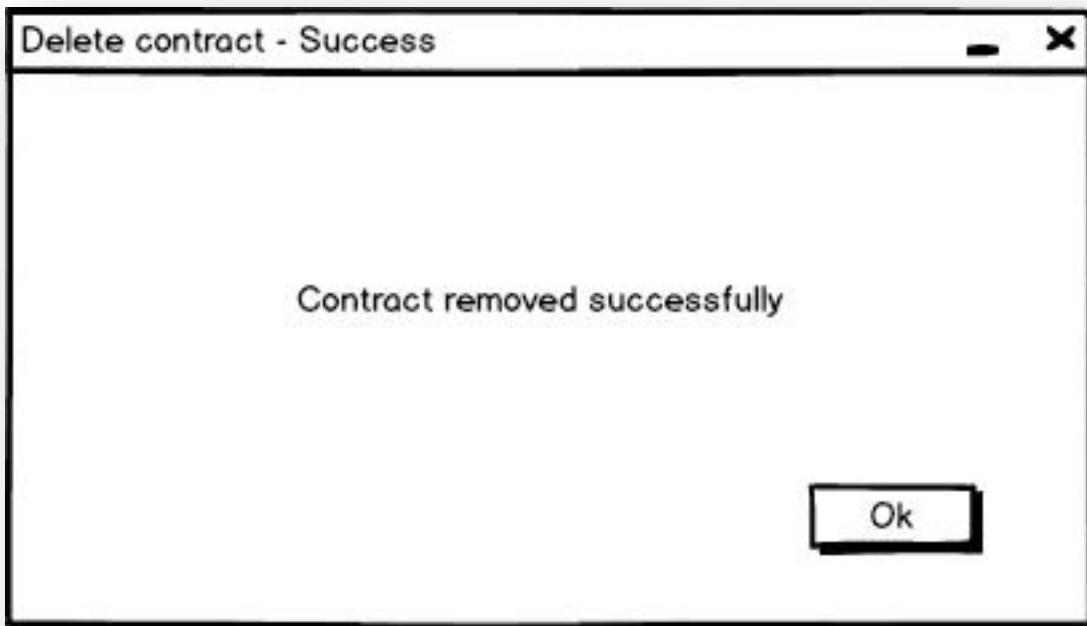
Delete contract



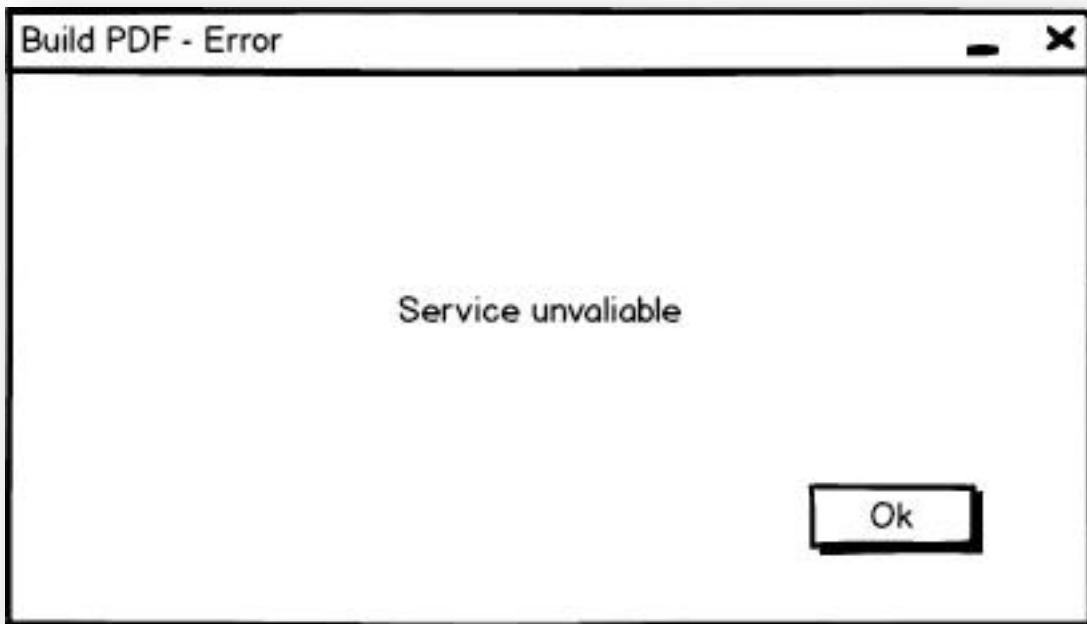
Delete contract - error



Delete injunction - error



Delete contract - success

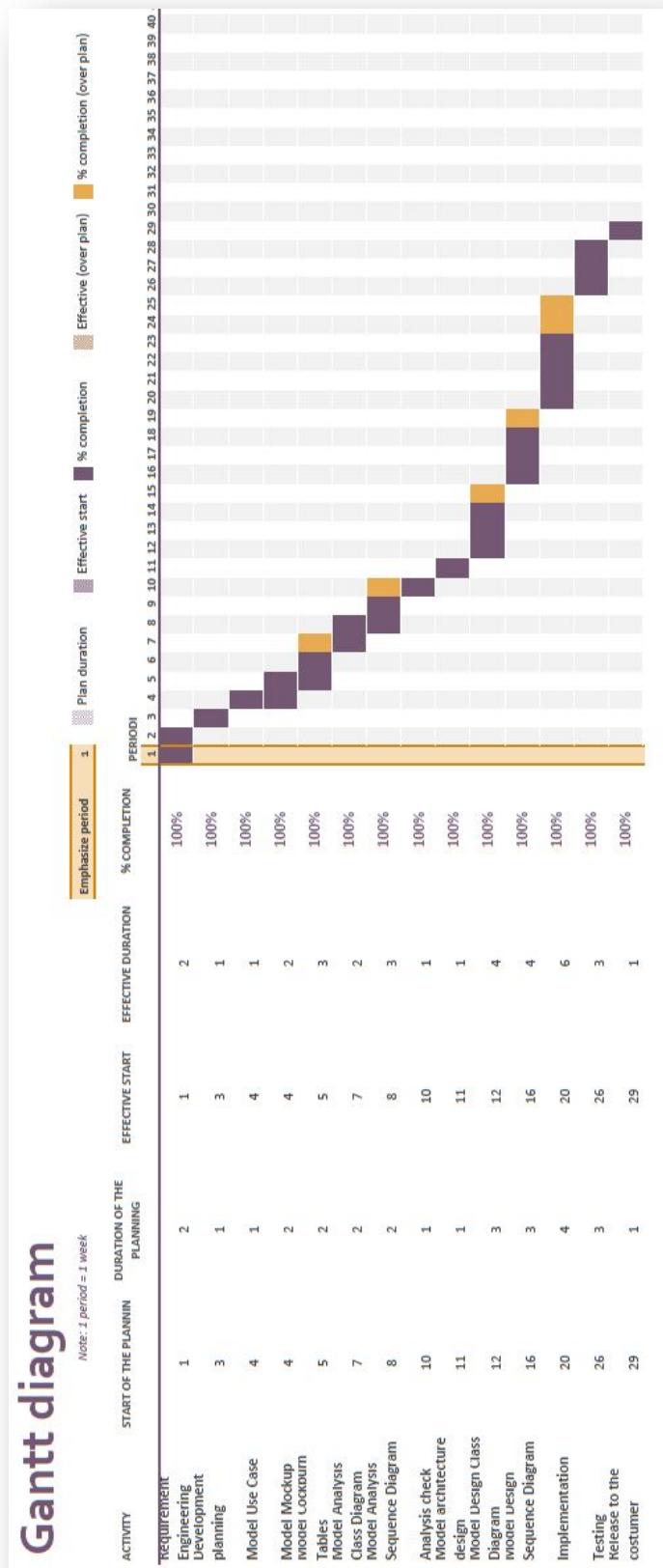


Build PDF - error



Delete injunctions – success

Gantt diagram



*Domain model**CRC Cards*

| | |
|--|----------------------|
| Class Name: AddContract_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows operation to add a new contract | Contract_MYSQL |
| Shows the “Add Contract” window | Contract |
| Builds and destroys the boundary objects needed for processing | Database_Controller |

| | |
|--|----------------------|
| Class Name: AlterContract_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows to modify an existing contract | Contract_MYSQL |
| Shows the “Alter Contract” window | Contract |
| Builds and destroys the boundary objects needed for processing | Database_Controller |

| | |
|--|----------------------|
| Class Name: RemoveContract_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows to delete an existing contract | Contract_MYSQL |
| Shows a popup to confirm the operation | Contract |
| | Database_Controller |

| | |
|--|----------------------|
| Class Name: SearchContract_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows to search an existing contract | Contract_MYSQL |
| Fills the table with the results | Contract |
| | Database_Controller |

| | |
|--|----------------------|
| Class Name: BillsQueue_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows the operator to start the process to confirm a bill | Bill_MYSQL |
| Fills the table with the bills that should be send | Bill |
| Shows the “BillsQueue” panel | Database_Controller |

| | |
|--|----------------------|
| Class Name: BillsHistory_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows the operator to view the bills history for a contract. | Database_Controller |
| Allows the operator to start the process to confirm a bill | Bill_MYSQL |
| Shows the “BillsHistory” window | Bill |
| Builds and destroys the boundary objects needed for processing | |

| | |
|--|----------------------|
| Class Name: ConfirmBill_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows the operator to confirm a single or a set of bills | Database_Controller |
| Shows the “BuildPDF” window | Bill_MYSQL |
| Shows the “SendPDF” window | Bill |
| Builds and destroys the boundary objects needed for processing | PDFMaker |
| | EMailSender |

| | |
|--|----------------------|
| Class Name: ConfirmInjunction_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows the operator to confirm an injunction | Database_Controller |
| Shows the “BuildPDF” window | Injunction_MYSQL |
| Shows the “SendPDF” window | Injunction |
| Builds and destroys the boundary objects needed for processing | PDFMaker |
| | EMailSender |

| | |
|---|----------------------|
| Class Name: InjunctionsHistory_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows the operator to view the injunctions history for a contract. | Database_Controller |
| Allows the operator to start the process to confirm an injunction | Bill_MYSQL |
| Shows the “InjunctionHistory” window | Injunction |
| Builds and destroys the boundary objects needed for processing | |

| | |
|---|----------------------|
| Class Name: InjunctionsQueue_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows the operator to start the process to confirm an injunction | Injunction_MYSQL |
| Fills the table with the bills that should be send | Injunction |
| Shows the “InjunctionsQueue” panel | Database_Controller |

| | |
|--|----------------------|
| Class Name: RemoveInjunction_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows to delete an existing contract | Injunction_MYSQL |
| Shows a popup to confirm the operation | Injunction |
| | Database_Controller |

| | |
|--|----------------------|
| Class Name: Login_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| It allows the operator to proceed to HomePage. | Operator_MYSQL |
| It controls the elaboration during the login. | Operator |
| Builds and destroys the boundary objects needed for processing | Database_Controller |

| | |
|--|----------------------|
| Class Name: Database_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| It allows connection to the Database | |
| It allows to execute the query on Database | |

| | |
|--|----------------------|
| Class Name: Listener | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| It allows to manage and interpret operator's actions | |

| | |
|--|----------------------|
| Class Name: Log_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Writes within a file the registration and the history of the operations performed by an operator or administrator. | Operator |
| Reports important events | |

| | |
|--|----------------------|
| Class Name: Main_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Shows the "Home" window | Operator |
| It allows the operator to proceed to logout | Database_Controller |
| It allows the operator to consult some help message for proper operation of the system | |



| | |
|--|----------------------|
| Class Name: Registry_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| It allows to direct the contract's crud operations | Contract |
| It allows to direct the display of bills history and injunctions history | Operator |

| | |
|--|----------------------|
| Class Name: ReportError_Controller | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| Allows operation to report error | Error MYSQL |
| Shows the "ReportError" window | Bill |
| Builds and destroys the boundary objects needed for processing | Database Controller |

| | |
|--|----------------------|
| Class Name: Bill | |
| Superclasses: Document | |
| Subclasses | |
| Responsibilities | Collaborators |
| It represents Bill entity | |
| It contains and provides bill's datas. | |

| | |
|---|----------------------|
| Class Name: Contract | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| It represents Contract entity | |
| It contains and provides contractr's datas. | |

| | |
|---|----------------------|
| Class Name: Document | |
| Superclasses | |
| Subclasses: Bill, Injunction | |
| Responsibilities | Collaborators |
| It represents Document entity | |
| It contains and provides documents's datas. | |

| | |
|--------------------------------|----------------------|
| Class Name: EMailSender | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| It allows to send Email | |

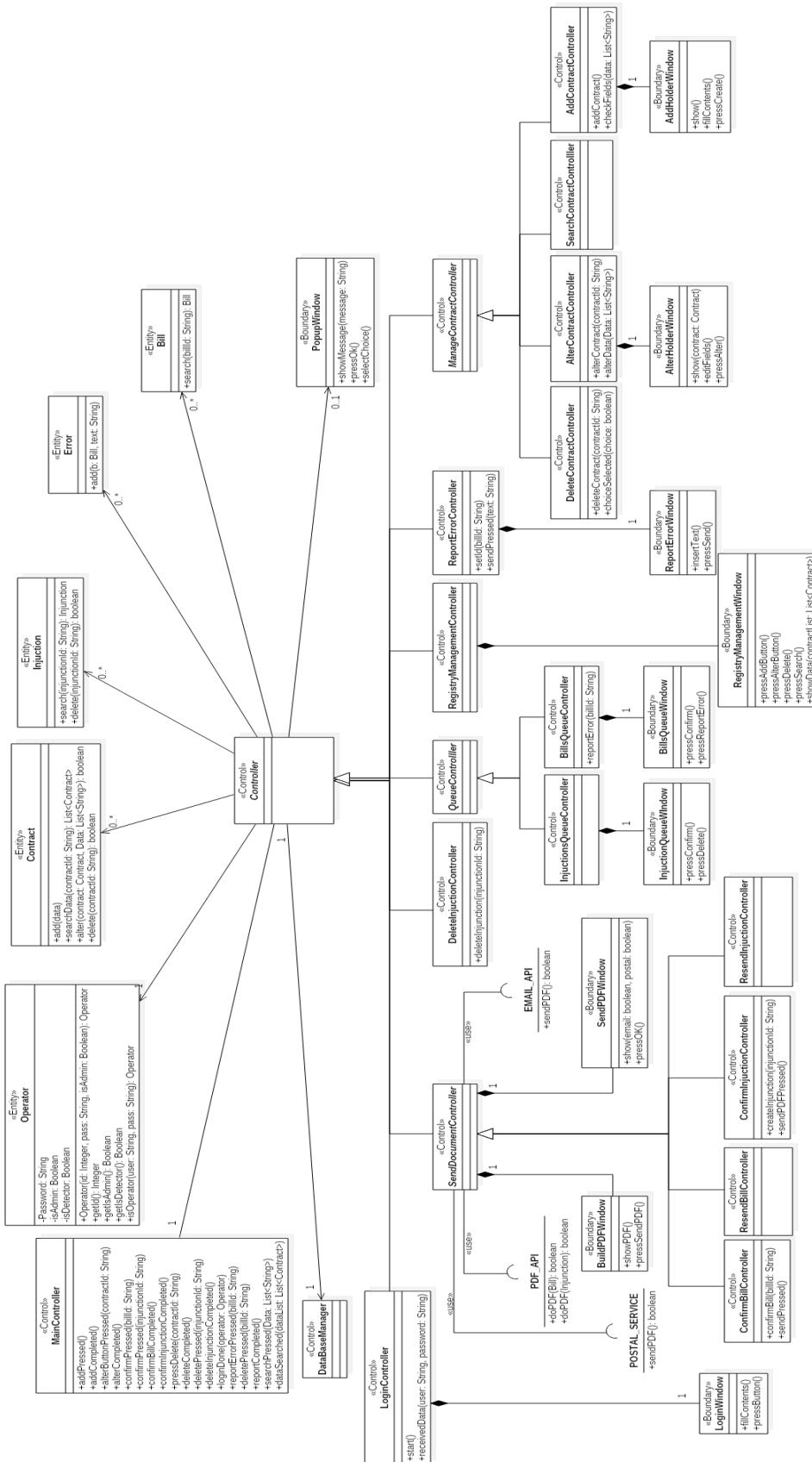
| | |
|---|----------------------|
| Class Name: ErrorModel | |
| Superclasses: Observable | |
| Subclasses | |
| Responsibilities | Collaborators |
| It represents Error entity | |
| It contains and provides error's datas. | |

| | |
|--|----------------------|
| Class Name: Injunction | |
| Superclasses: Document | |
| Subclasses | |
| Responsibilities | Collaborators |
| It represents Injunction entity | |
| It contains and provides injunction's datas. | |

| | |
|---|----------------------|
| Class Name: Operator | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| It represents Operator entity | |
| It contains and provides operator's datas | |

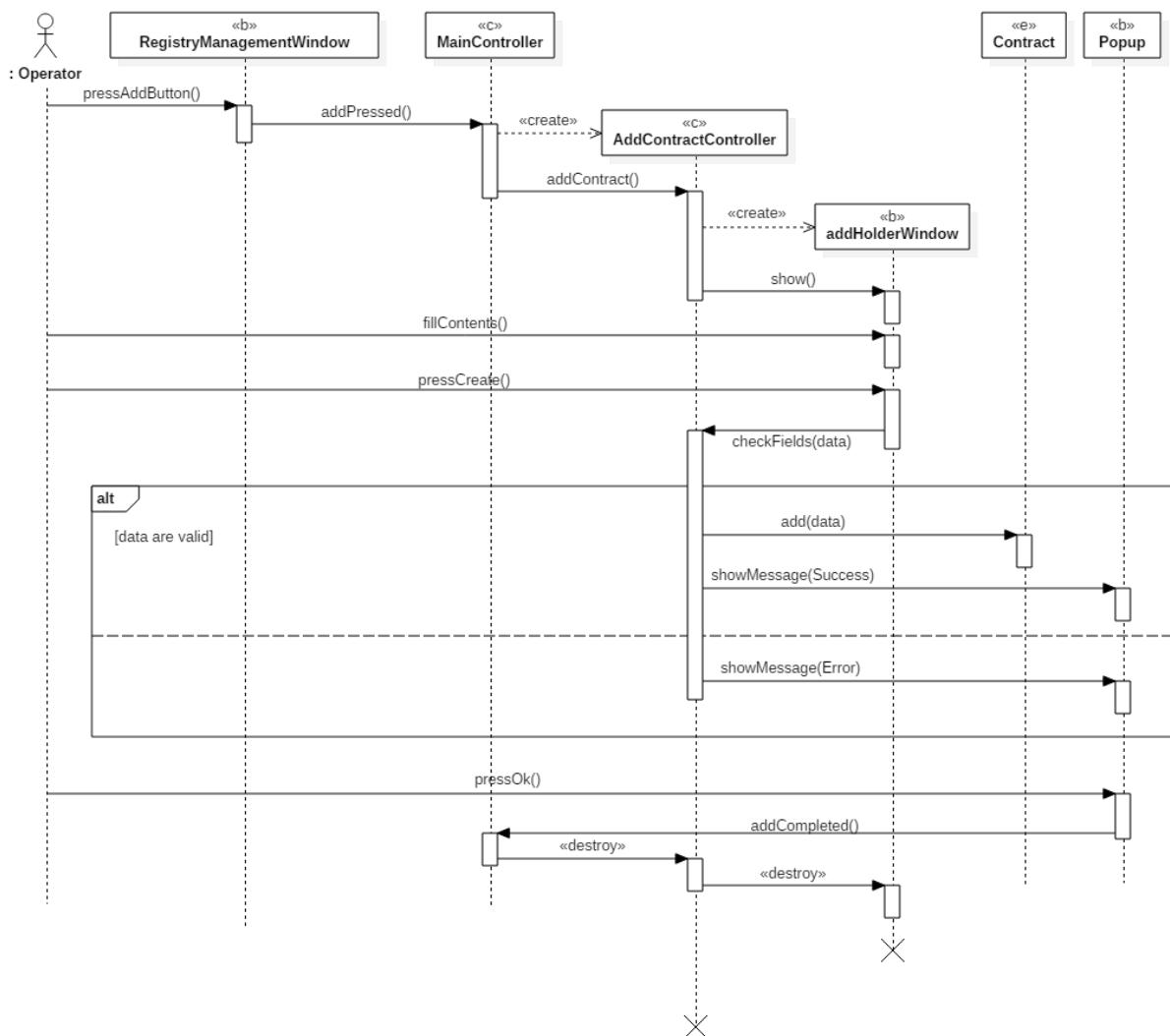
| | |
|-----------------------------|----------------------|
| Class Name: PDFMaker | |
| Superclasses | |
| Subclasses | |
| Responsibilities | Collaborators |
| It allows to create a PDF | |

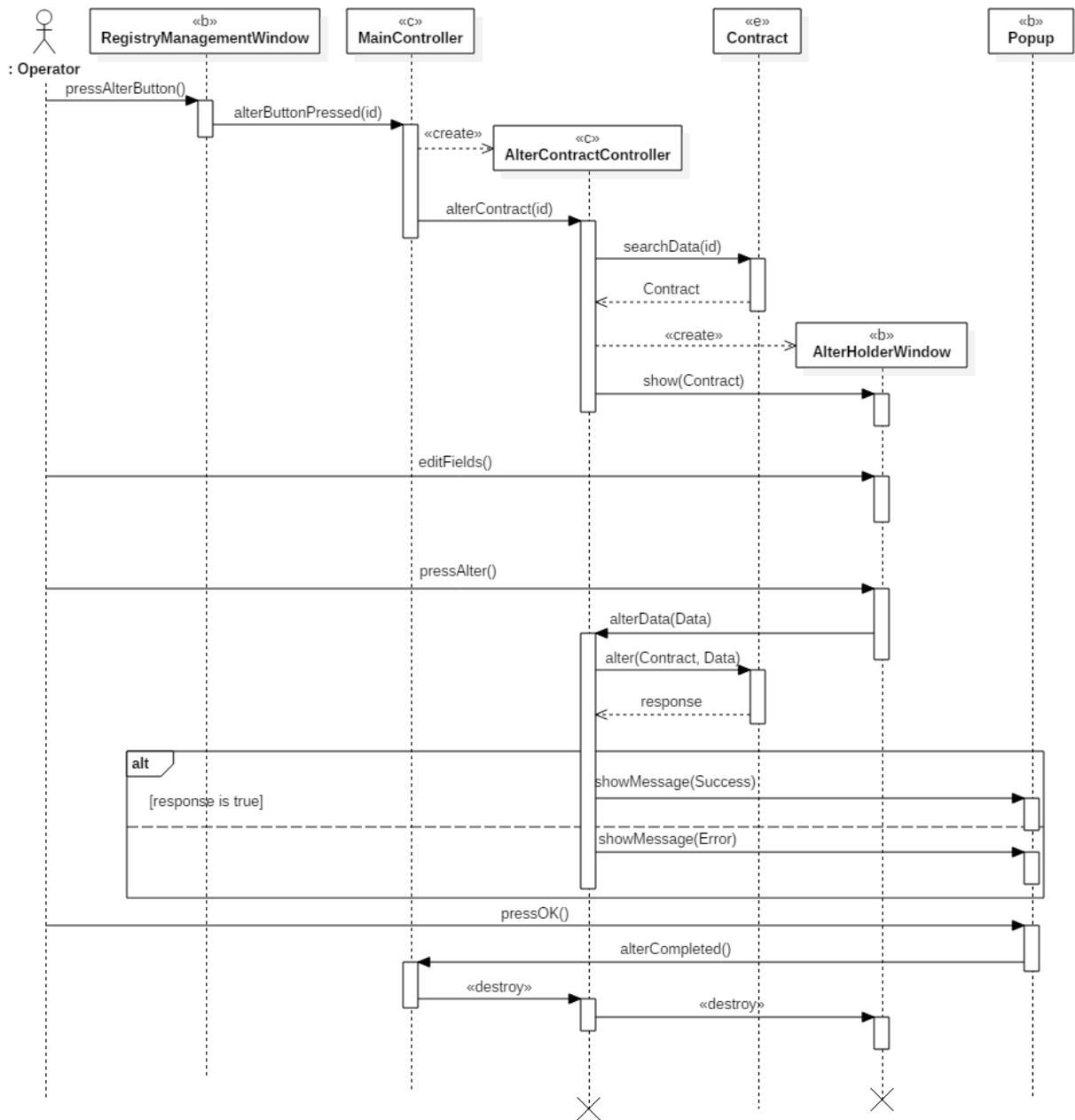
Analysis class diagram

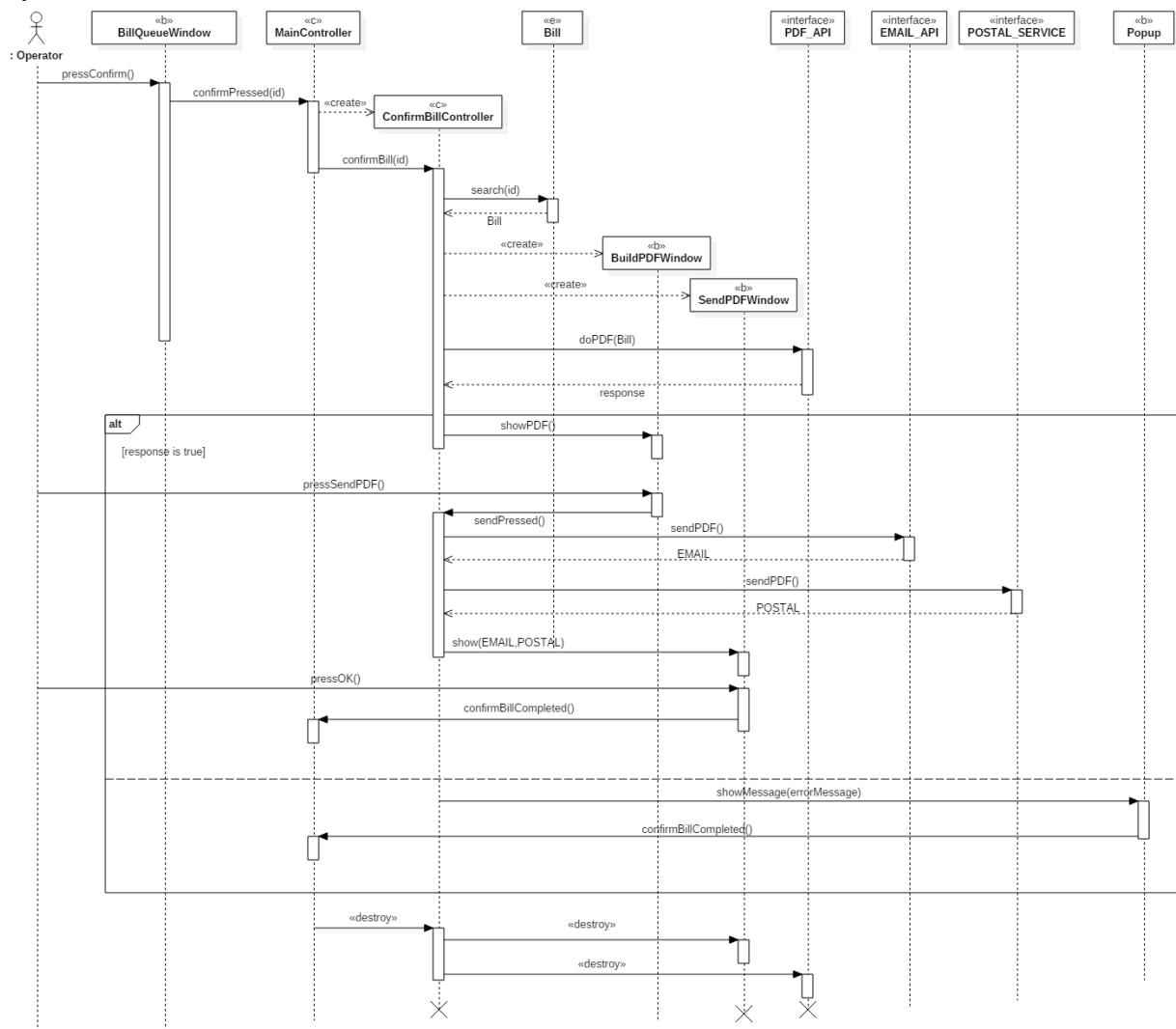


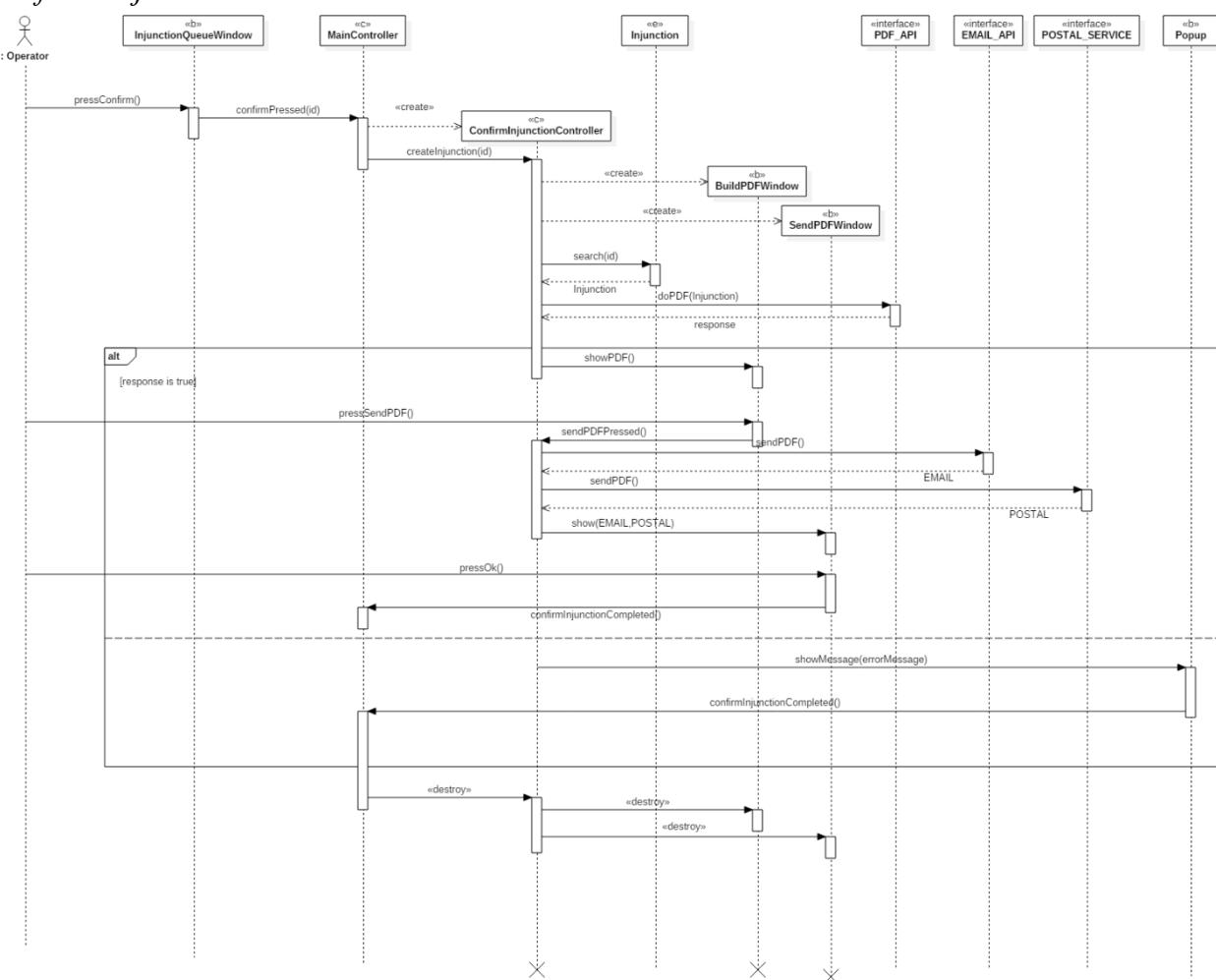
Analysis sequence diagrams

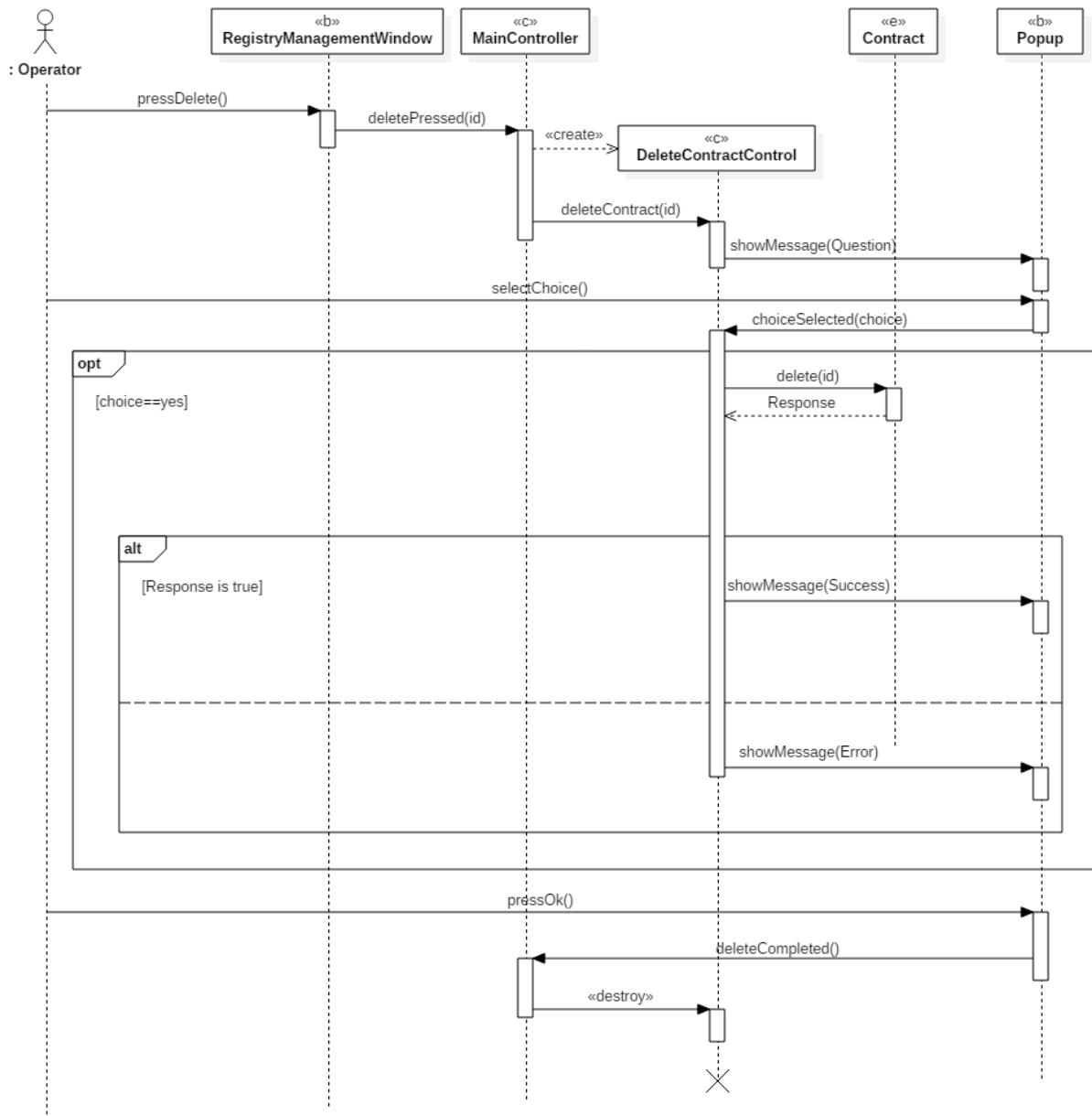
Adds contract

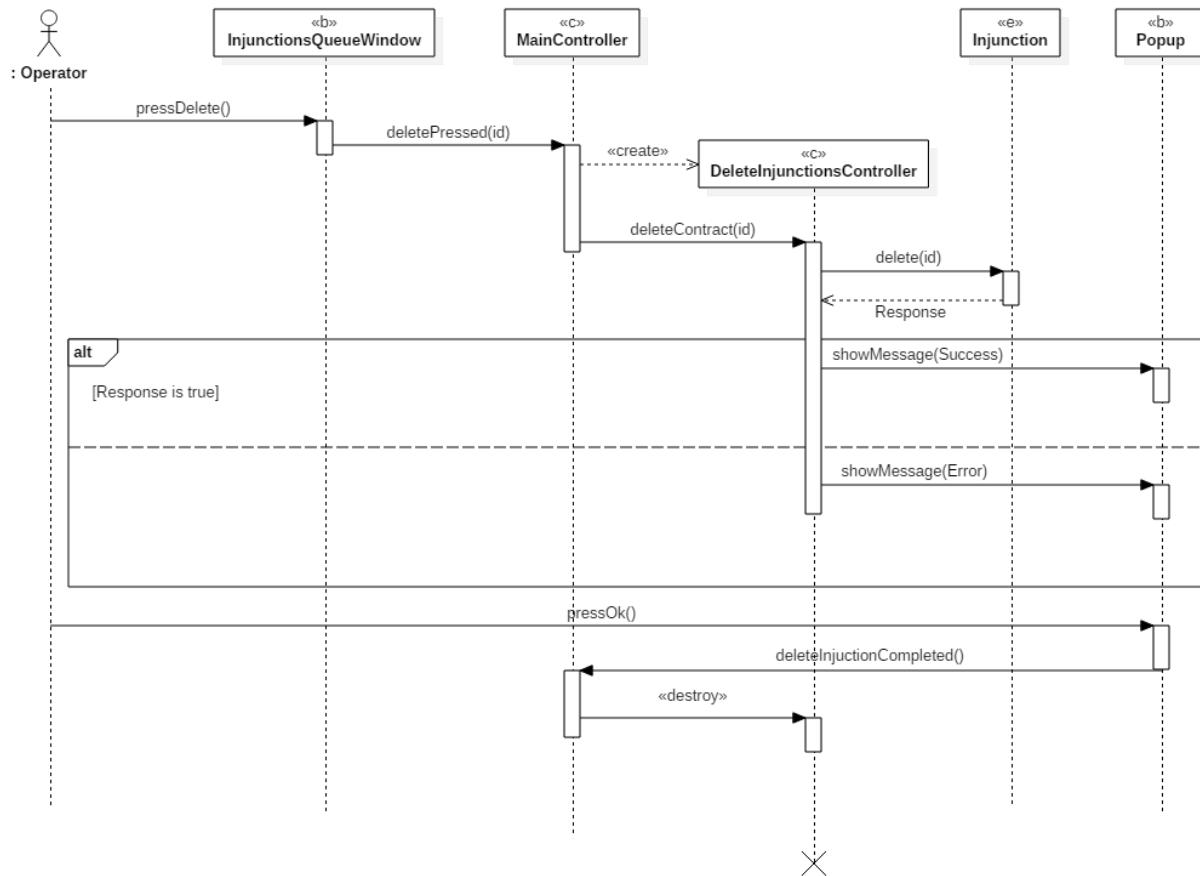


Alters contract

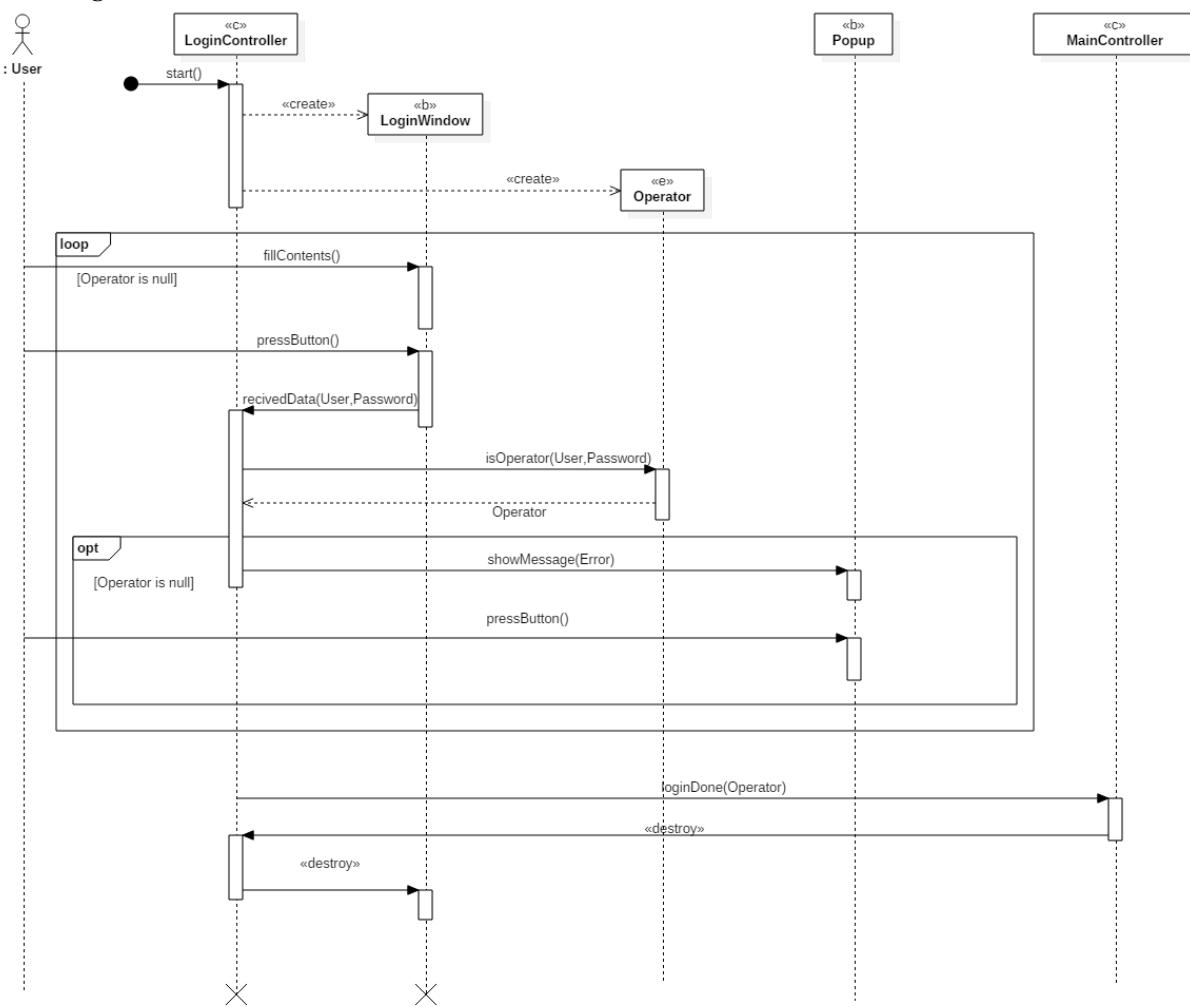
Confirms bill

Confirms injunctions

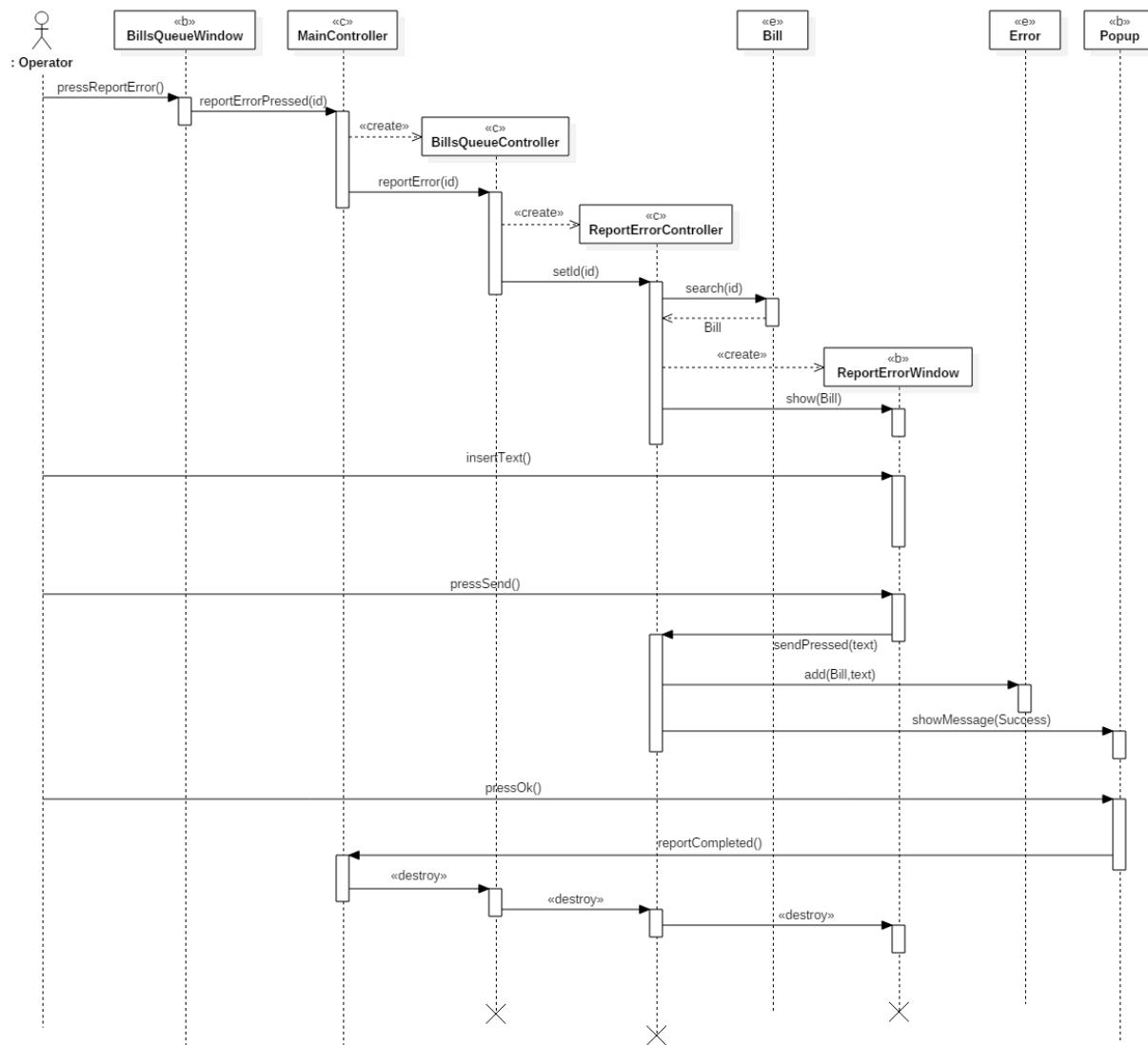
Deletes contract

Deletes injunctions

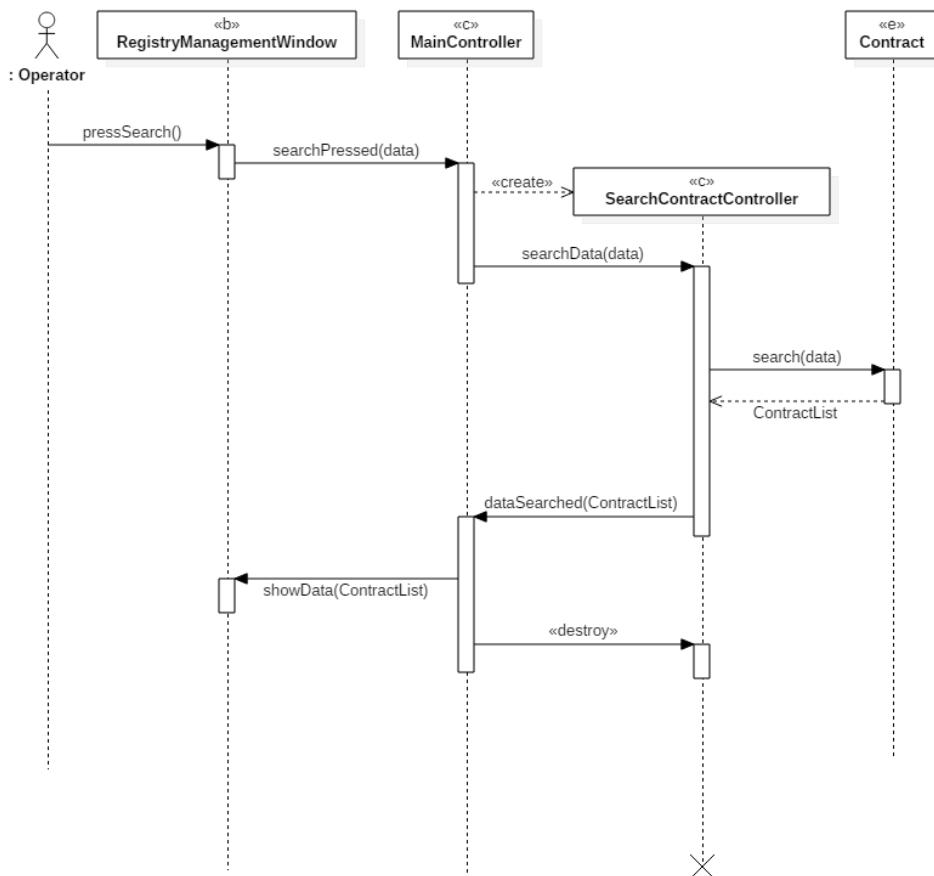
Performs logi



Reports errors in bills



Researches contract



System design document

Architecture analysis

The software architecture used in the application is the Model-View-Controller architecture.

The system is partitioned into three subsystems:

- The Model subsystem, it maintains the application domain knowledge
- The View subsystem, it allows the user to view the object of the application domain
- The Control subsystem, it deals with the user interactions

A set of data in a relational database, used by the application, and the entity classes are part of the first subsystem (they represent the entities of the problem).

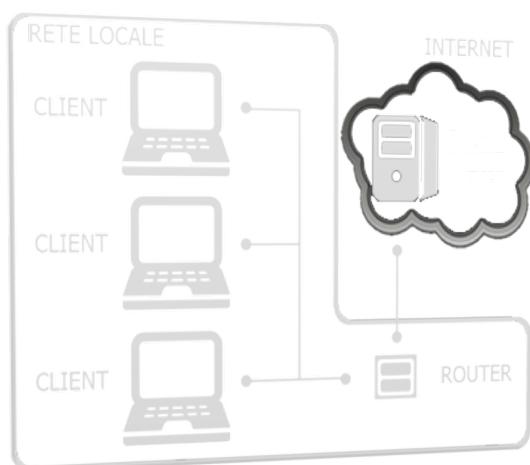
The View subsystem is formed by the boundary classes. They form the user interfaces.

The Control subsystem is formed by the control classes. They manage the control logic and the interaction between entity and user interfaces.

The application manages the asynchronous event by an event-based management. The boundary objects, that form the user interfaces, are associated to listeners that catch an event when it occurs and execute actions depending by event.

The application is developed using centralized control: only one subsystem, the Control subsystem, is responsible for activating and stopping the others.

To have a centralized database the application is based on a cloud solution. With a database as a service model, application owners do not have to install and maintain the database themselves. Instead, the database service provider takes responsibility for installing and maintaining the database, and application owners are charged according to their usage of the service.



To give access to operators' log file to system administrator, the application uploads all files on a cloud ftp server.

Technologies

- **Apache PDFBox:** “*The Apache PDFBox® library is an open source Java tool for working with PDF documents. This project allows creation of new PDF documents, manipulation of existing documents and the ability to extract content from documents. Apache PDFBox also includes several command-line utilities. Apache PDFBox is published under the Apache License v2.0.*” (<https://pdfbox.apache.org/>)
- **ICEpdf:** “*ICEpdf is an open source PDF engine for viewing, printing, and annotating PDF documents. The ICEpdf API is 100% Java, lightweight, fast, efficient, and very easy to use. ICEpdf can be used as standalone open source Java PDF viewer, or can be easily embedded in any Java application to seamlessly load or capture PDF documents.*” (<http://www.icesoft.org/java/projects/ICEpdf/overview.jsf>)
- **Apache J4LOG:** “*With log4j it is possible to enable logging at runtime without modifying the application binary. The log4j package is designed so that these statements can remain in shipped code without incurring a heavy performance cost. Logging behavior can be controlled by editing a configuration file, without touching the application binary.*” (<https://logging.apache.org/log4j/1.2/>)
- **Oracle JavaMail:** “*The JavaMail API provides a platform-independent and protocol-independent framework to build mail and messaging applications. The JavaMail API is available as an optional package for use with the [Java SE platform](#) and is also included in the [Java EE platform](#).*” (<http://www.oracle.com/technetwork/java/javamail/index.html>)
- **Apache Commons-net:** “*Apache Commons Net™ library implements the client side of many basic Internet protocols. The purpose of the library is to provide fundamental protocol access, not higher-level abstractions. Therefore, some of the design violates object-oriented design principles. Our philosophy is to make the global functionality of a protocol accessible (e.g., TFTP send file and receive file) when possible, but also provide access to the fundamental protocols where applicable so that the programmer may construct his own custom implementations (e.g, the TFTP packet classes and the TFTP packet send and receive methods are exposed).*” (<https://commons.apache.org/proper/commons-net/>)
- **MySQL Connector:** “*MySQL provides standards-based drivers for JDBC, ODBC, and .Net enabling developers to build database applications in their language of choice.*” (<https://www.mysql.com/it/products/connector/>)
- **GitHub:** “*GitHub is a web-based Git or version control repository and Internet hosting service. It is mostly used for code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project*” (<https://github.com>)

Design pattern & Implementation choices

The design pattern used are the Observer, Singleton and DAO pattern.

- The **Observer pattern** is used to define the Control - Component relationship. Each class that implements the Control interface has their Component objects; the way by which this classes dialogue is defined by the Observer Pattern. For increase the modularity has been included a class Listener. This last is used by Control classes (and by Main_Controller class) for catching the event triggered by their Component objects. After we talk about the class Listener.
- The **Singleton pattern** is used to define the classes Main_Controller and Database_Controller. The application need just one instance of both these classes. Using the Singleton pattern, we are sure that, in every moment during the computation, we have at most one instance of Main_Controller and one instance of Database_Controller. The reason to have only one Main_Controller is that this class manage the main control logic of the application: having more than one Main_Controller would be a disaster. Instead, the reason to have only one Database_Controller is that this class is the only one that dialogue with the database: it asks the database to do queries and it manages the results. We want the database receives a request at a time.
- The **DAO (Data Access Object) pattern** is used to provide some specific operation without exposing details of the database technology implemented. In this way the system is independent from the database technology.
Any DAO class is used to access to a specific object stored into the DBMS (all of these classes are described into the DAO Package).
In our case we used as technology the mySql DBMS.

During the design, some implementation choices were made:

1: Class Listener: it has been introduced to increase the modularity.

This class implements the ActionListener, MouseListener and ChangeListener interfaces, so it implements the methods described into the implemented interfaces.

These methods are invoked by the Component objects and it does something different depending on the object that invoked it.

2: Controller interface: it has been introduced as tag interface.

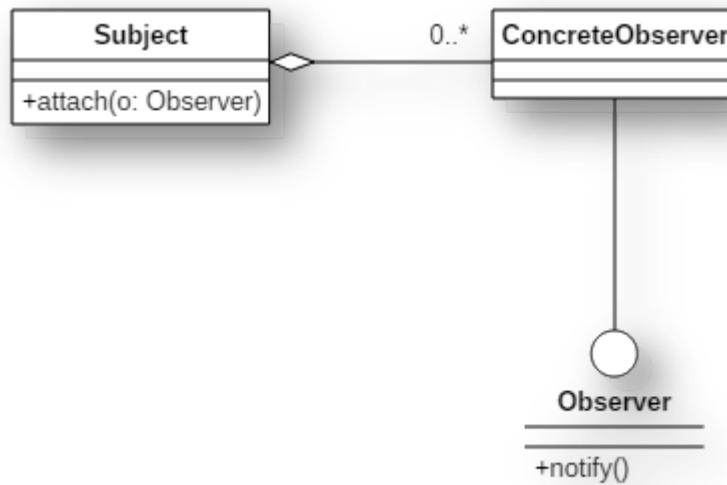
3: Classes Registry_Controller, BillsQueue_Controller, InjunctionQueue_Controller: Main_Controller's responsibilities (which is present into the Analysis' class diagram) were splitted, during the design phase, into these 3 classes: Registry_Controller, BillsQueue_Controller, InjunctionQueue_Controller. In this way we increased the modularit: each one of these 3 classes manage one of the 3 panels present into the main view (Registry Management, Bills Queue, Injunctions Queue).

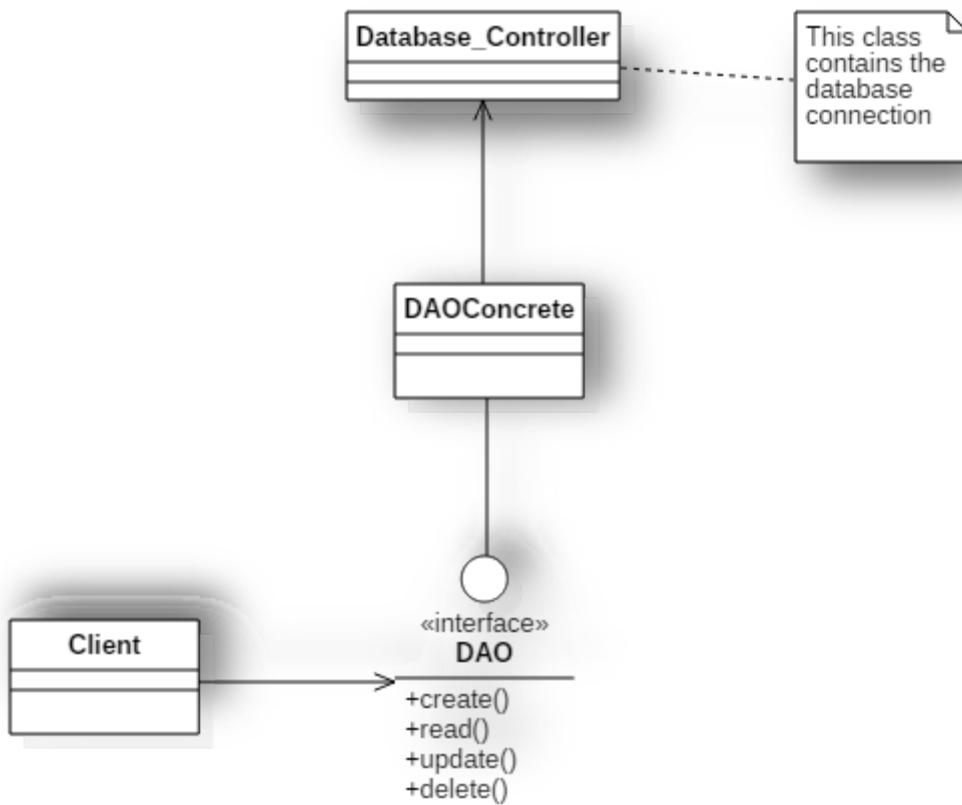
4: All log's file generated from the J4LOG API are shown on <http://loggci16.altervista.org/> (password "ingsw12a3v").

Singleton pattern:

| Singleton |
|-------------------------------|
| -singletonInstance: Singleton |
| -Singleton() |
| +getInstance(): Singleton |

Observer pattern

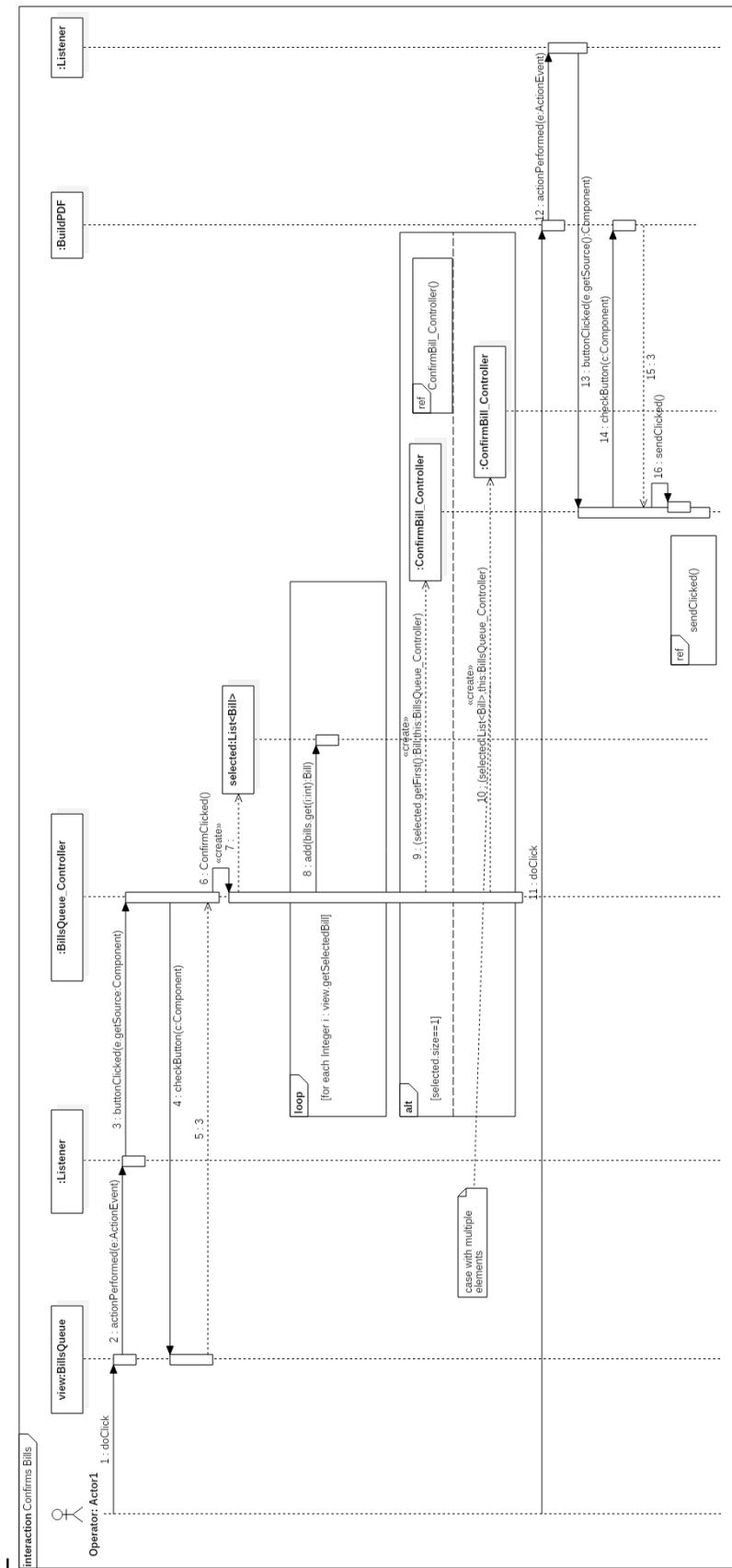


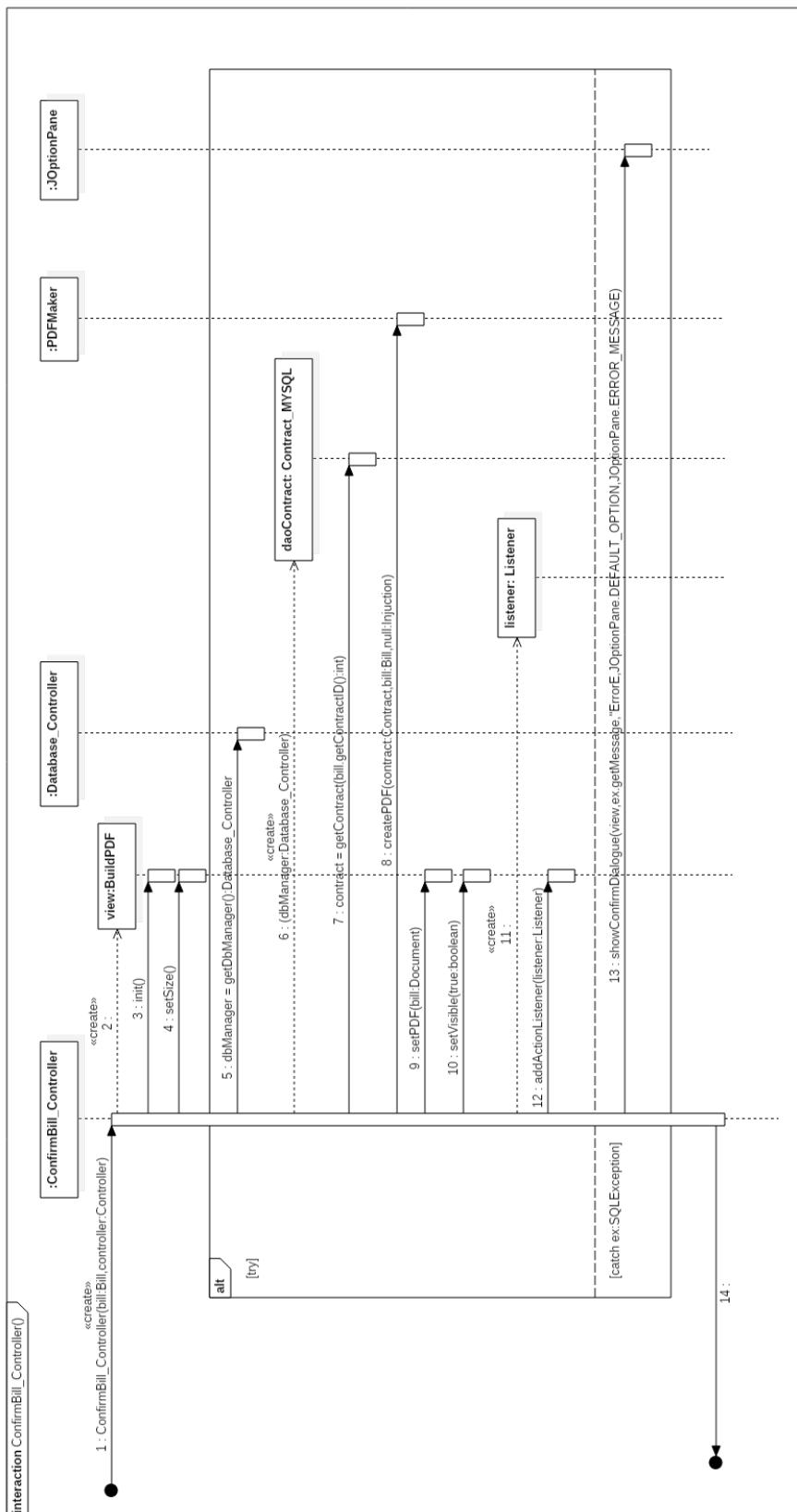
DAO pattern



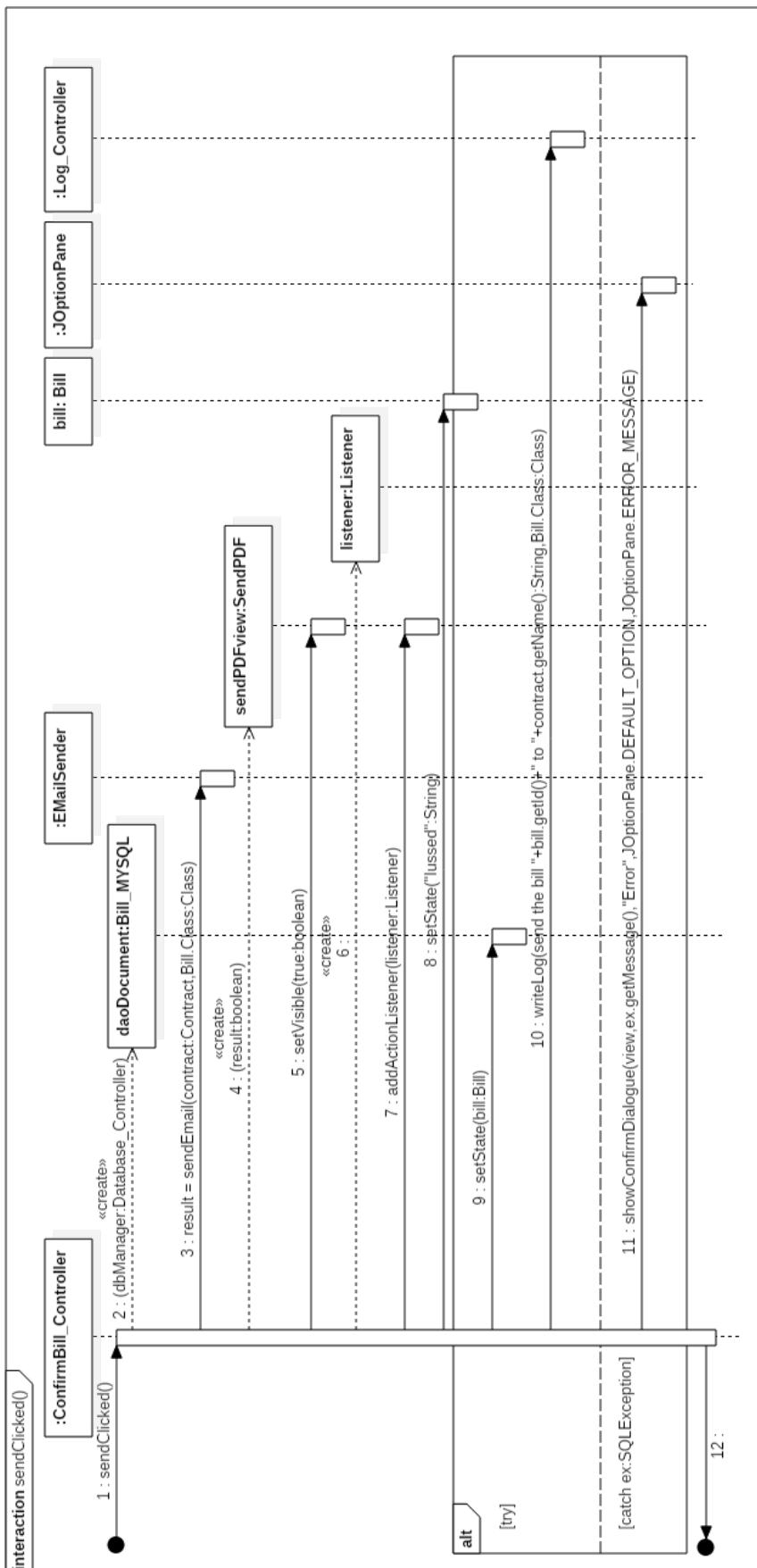
Design sequence diagram

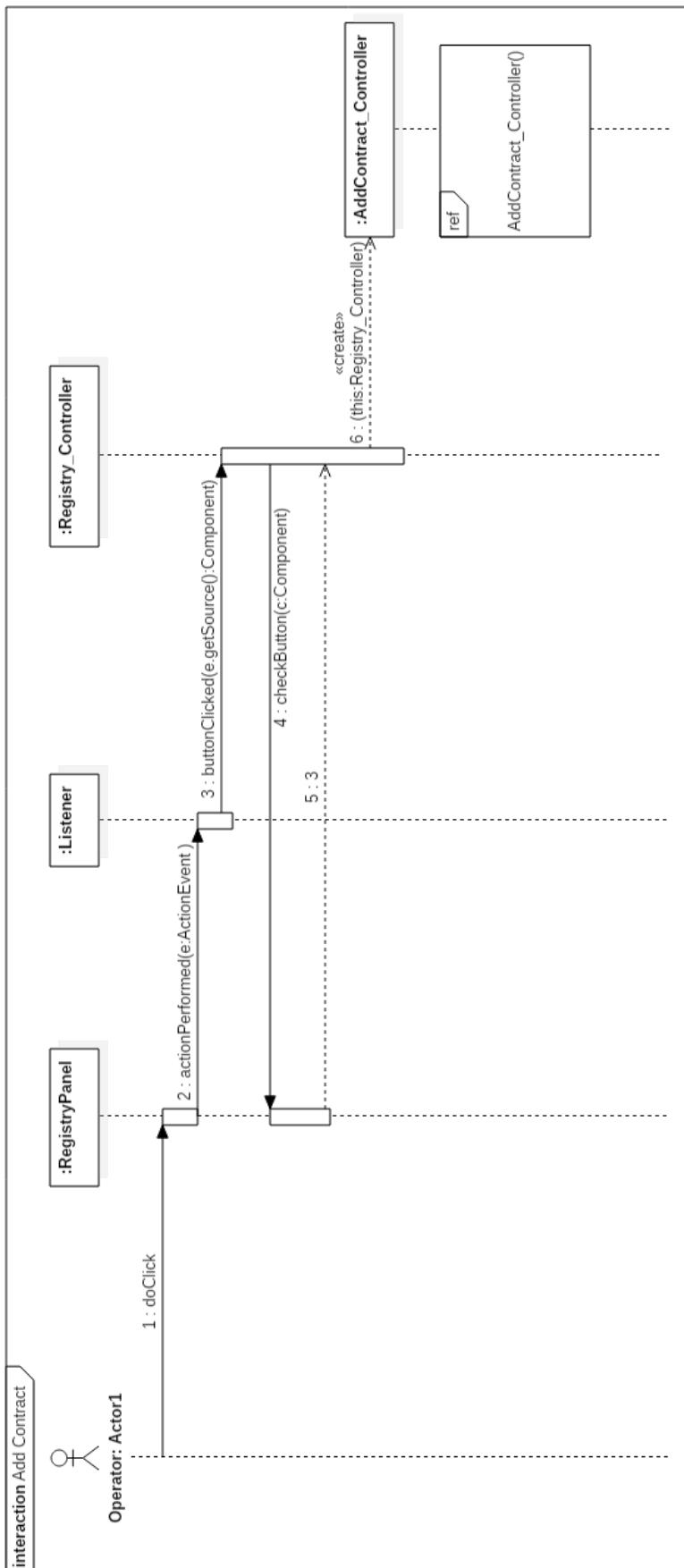
Confirm bill

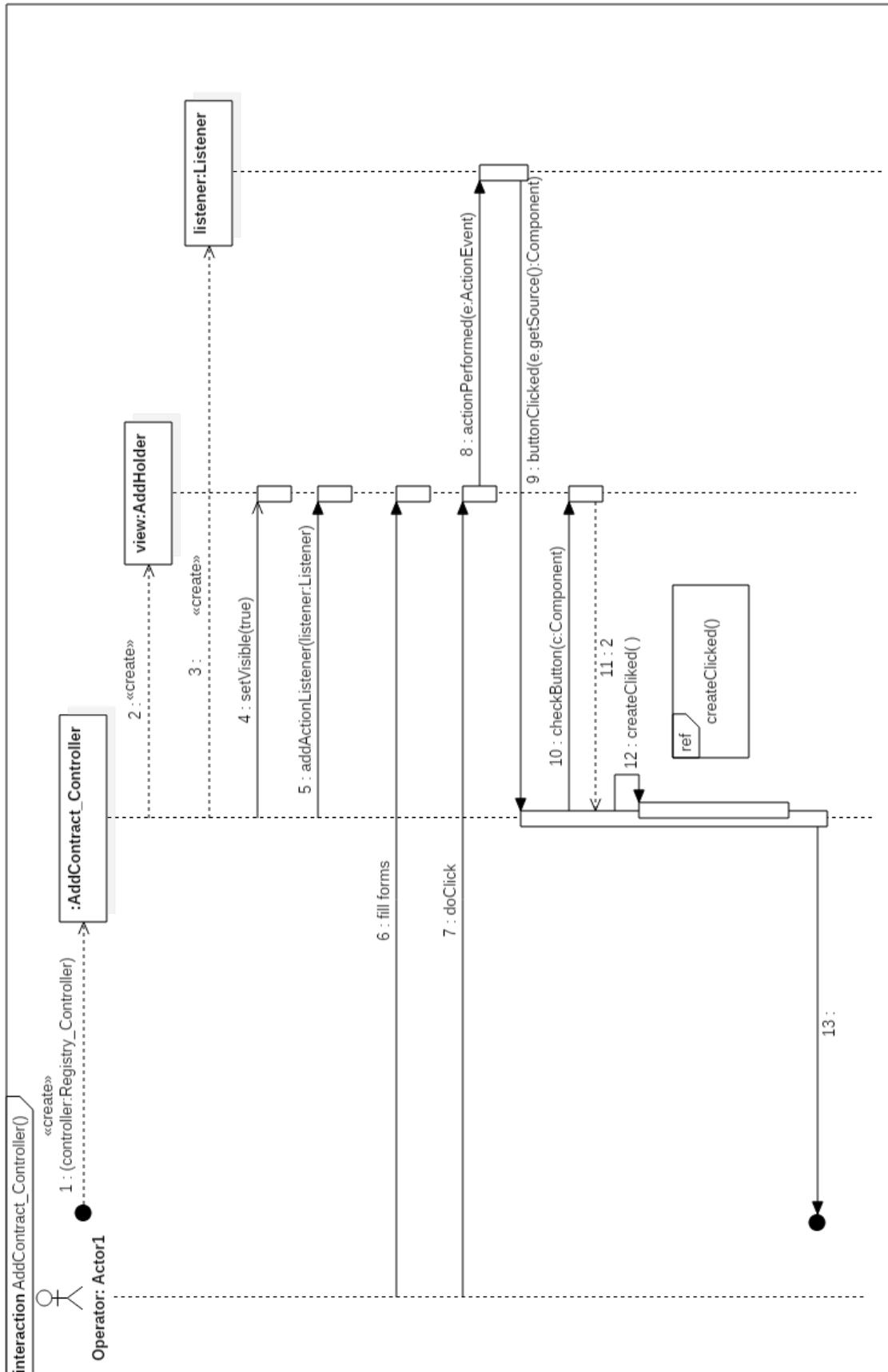


ConfirmBill_Controller()

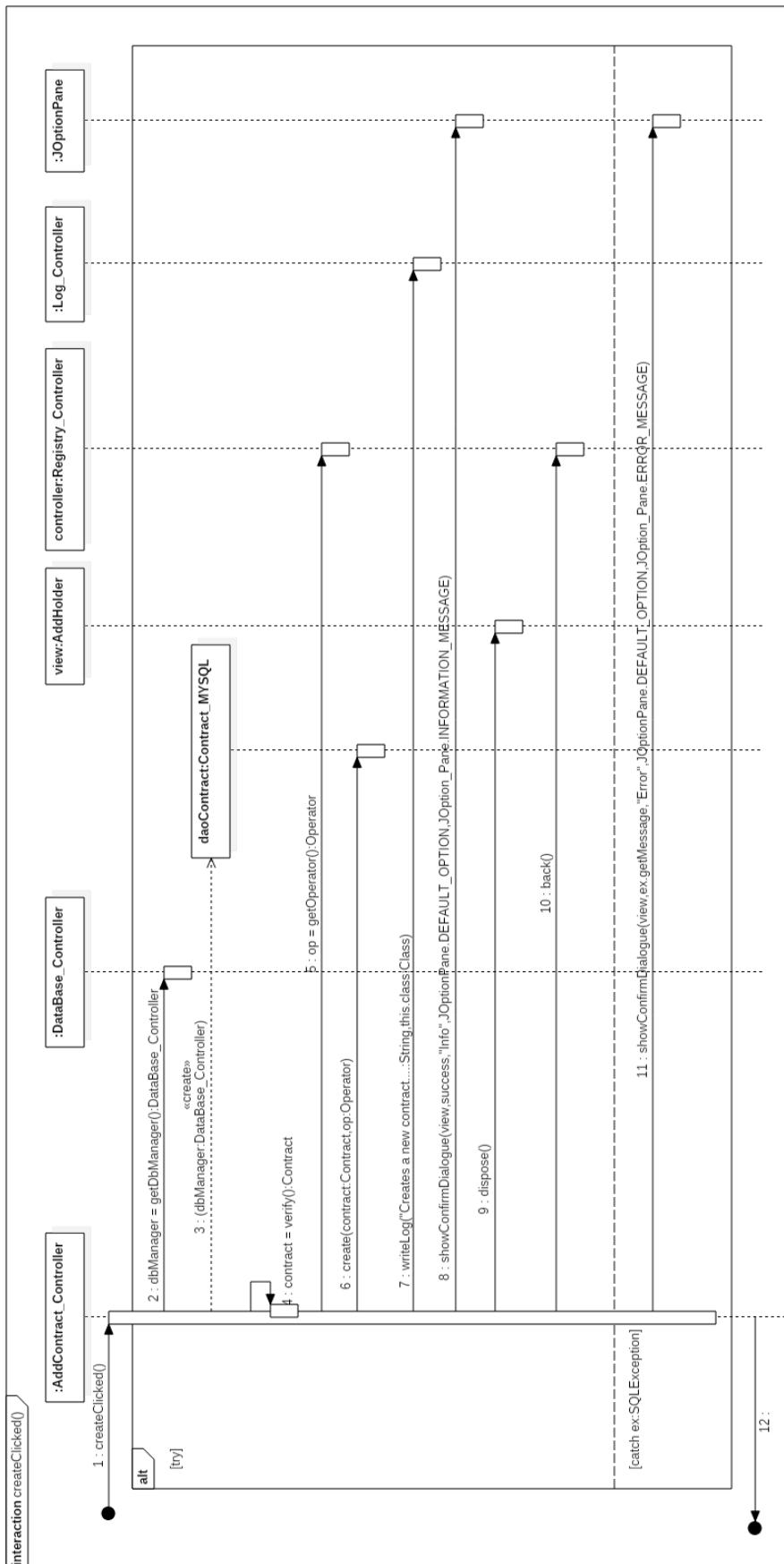
ConfirmsBill_Controller.sendClicked()



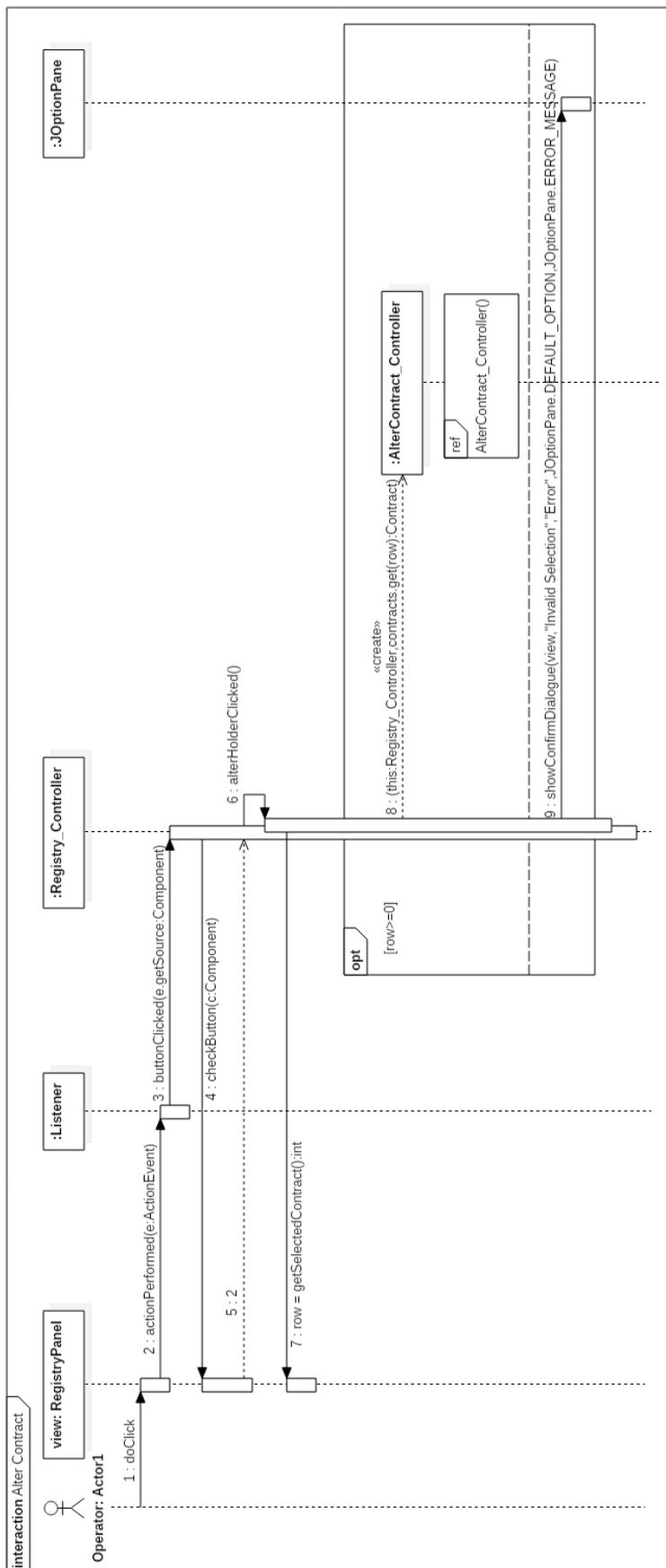
Add contract

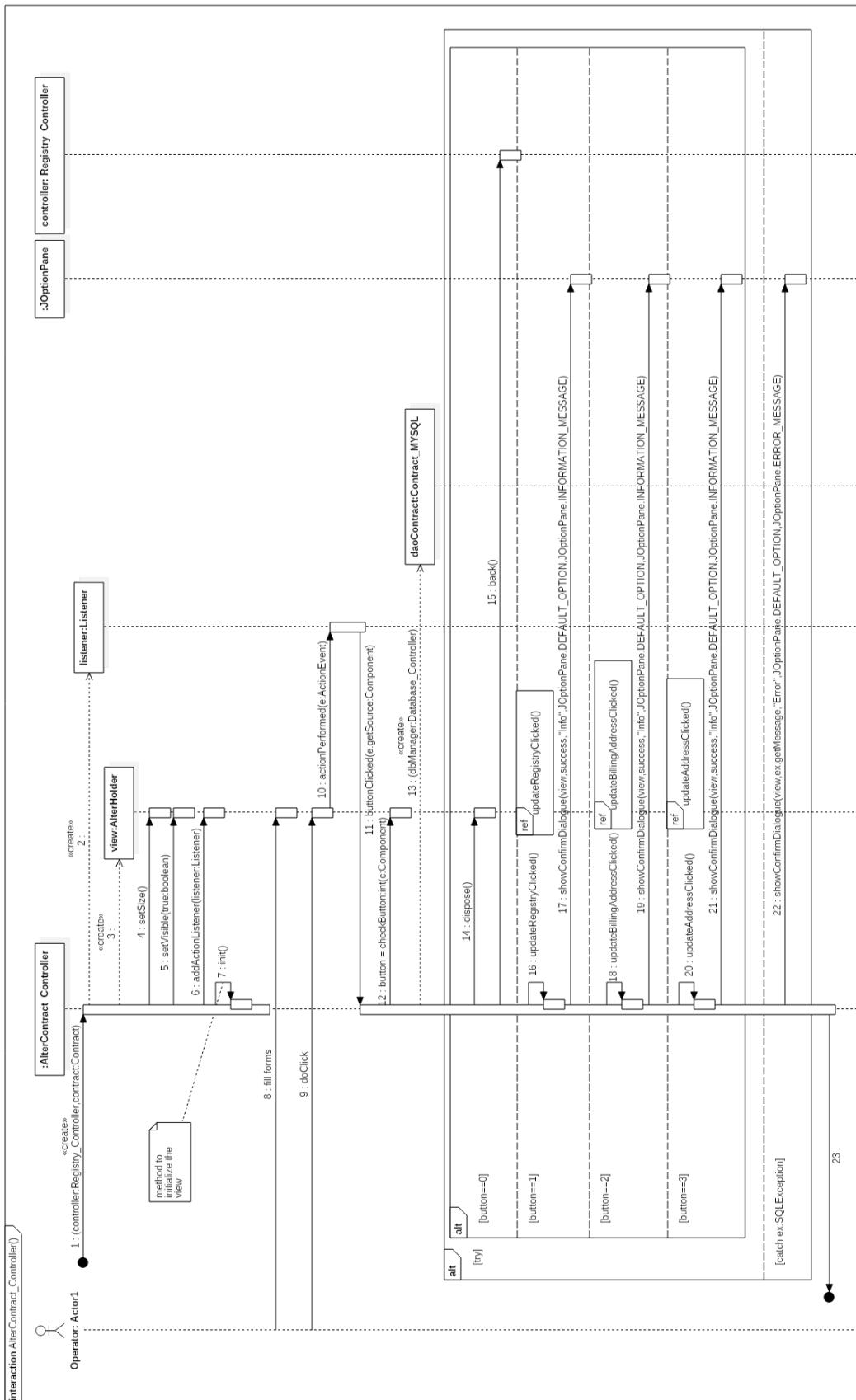
AddContract_Controller()

AddContract_Controller.createClicked()

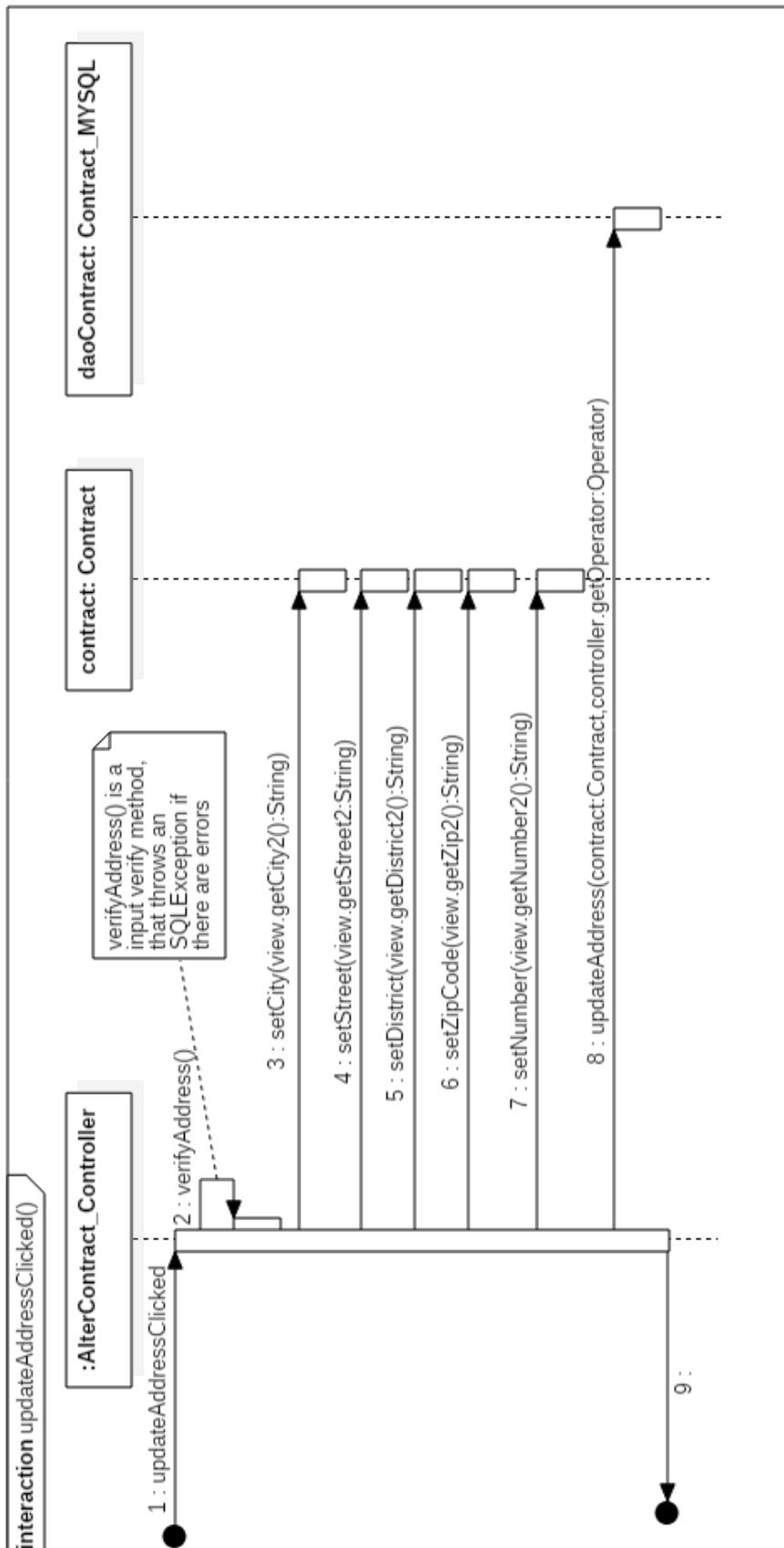


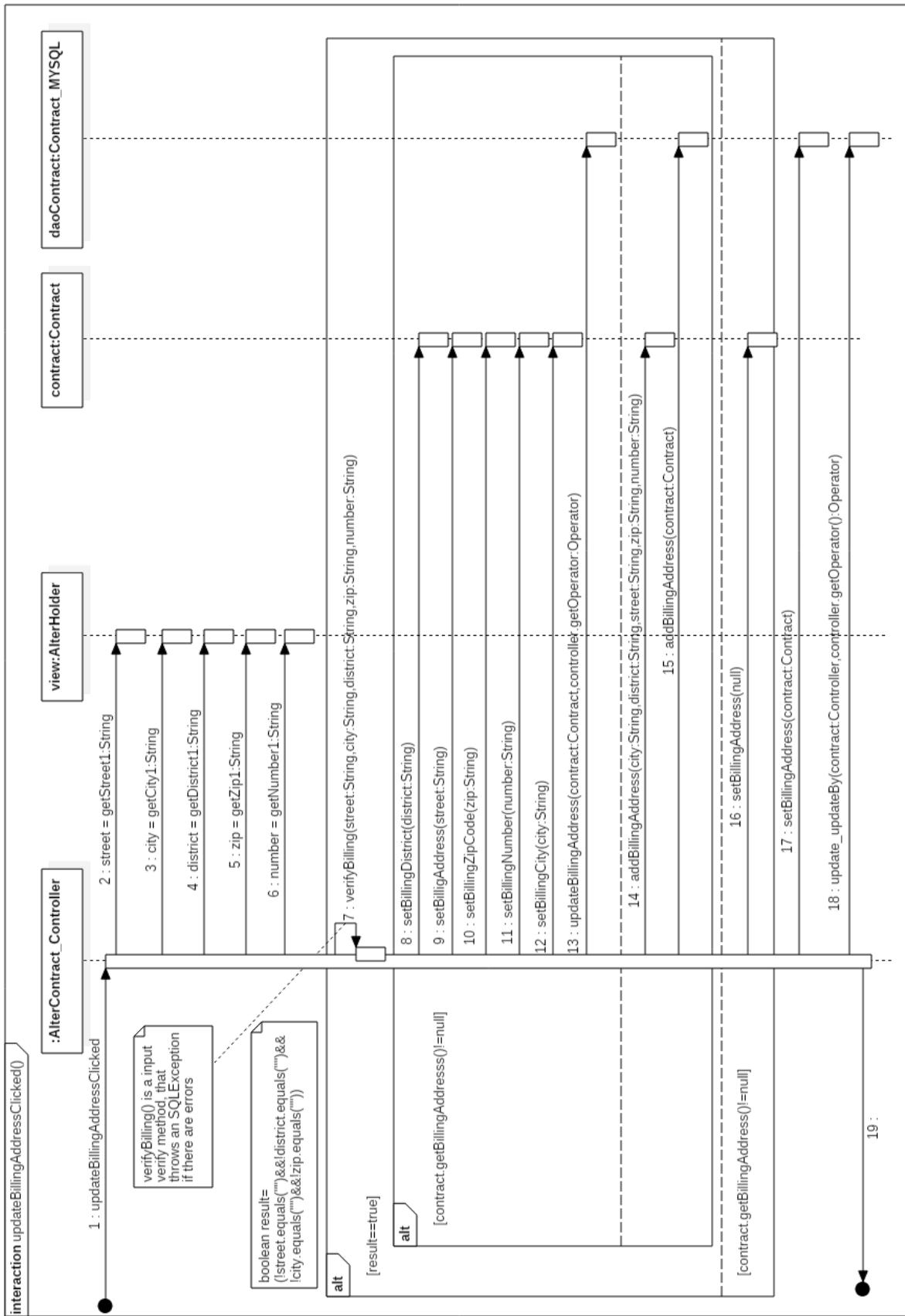
Alter contract



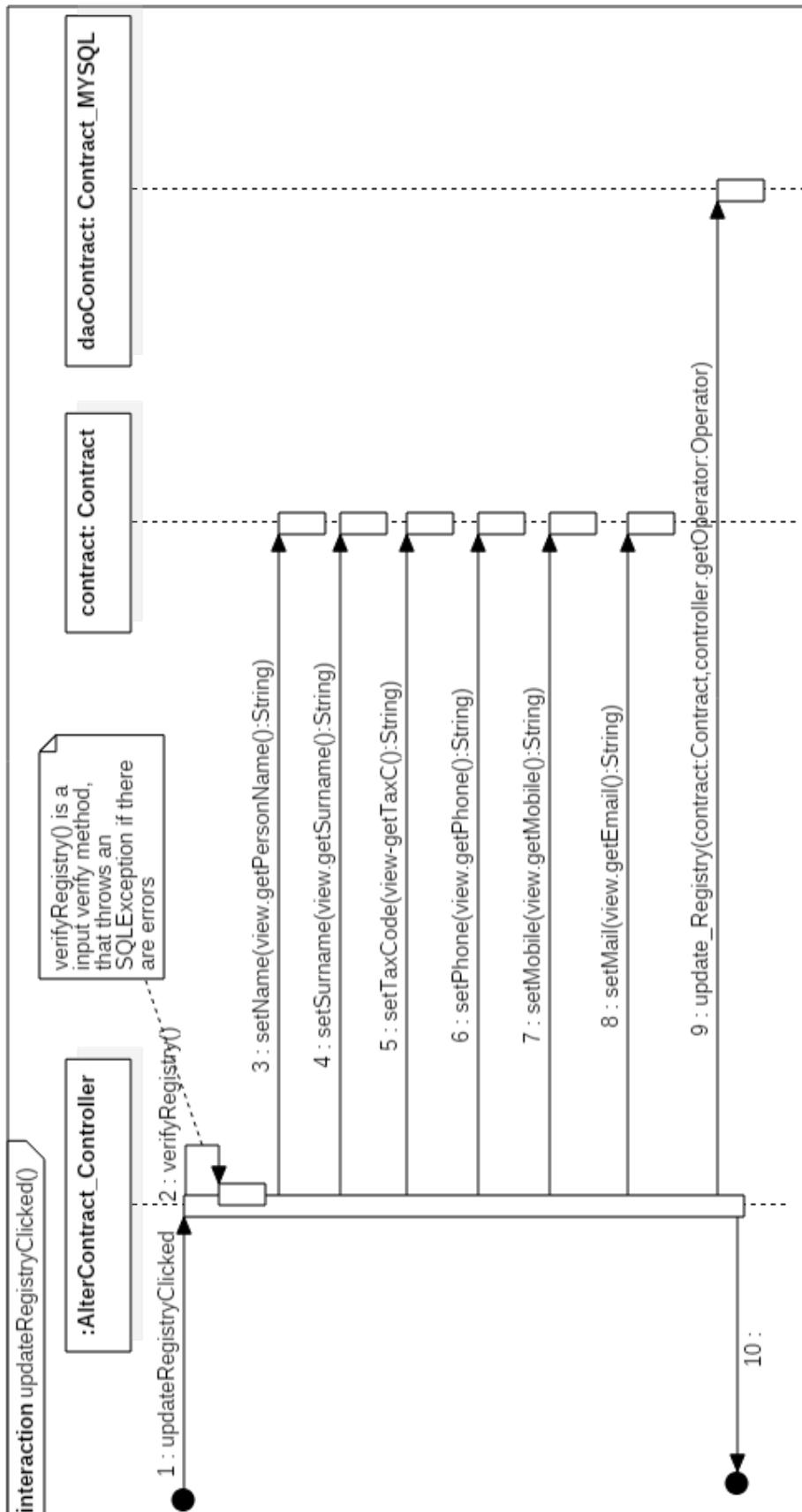
AlterContract_Controller()

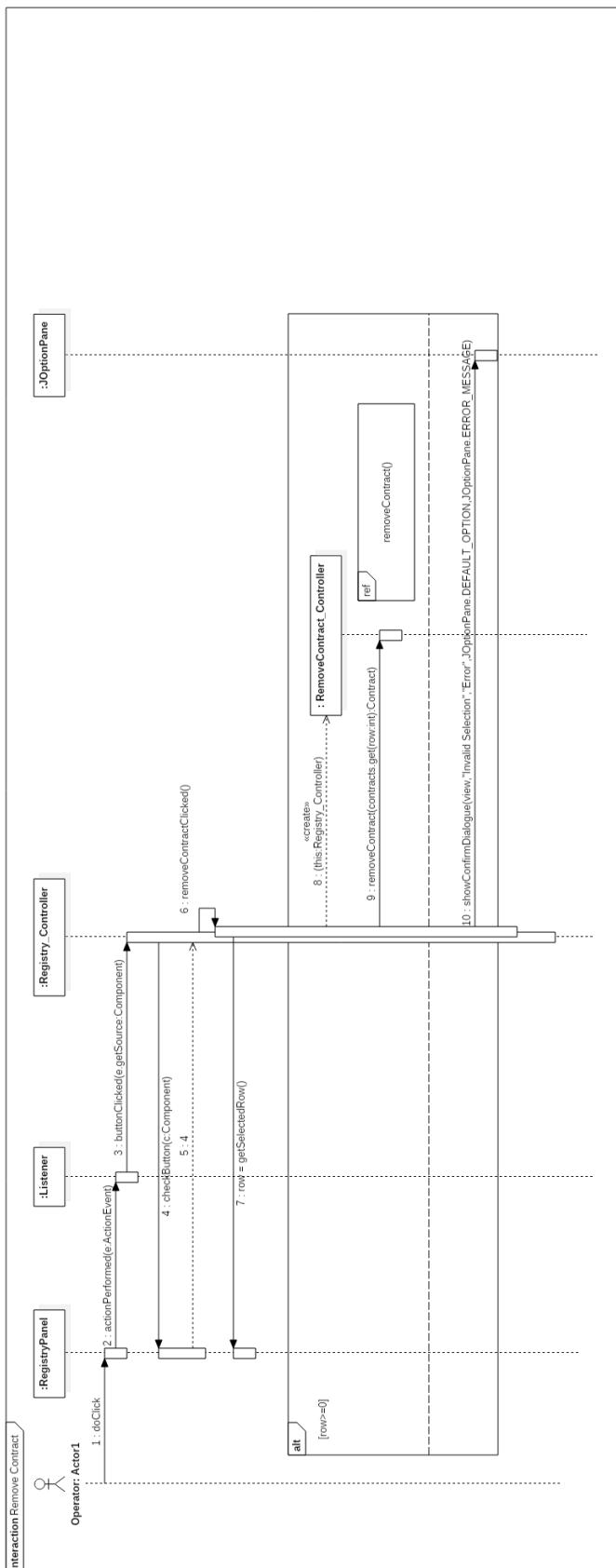
AlterContract_Controller.addressClicked()

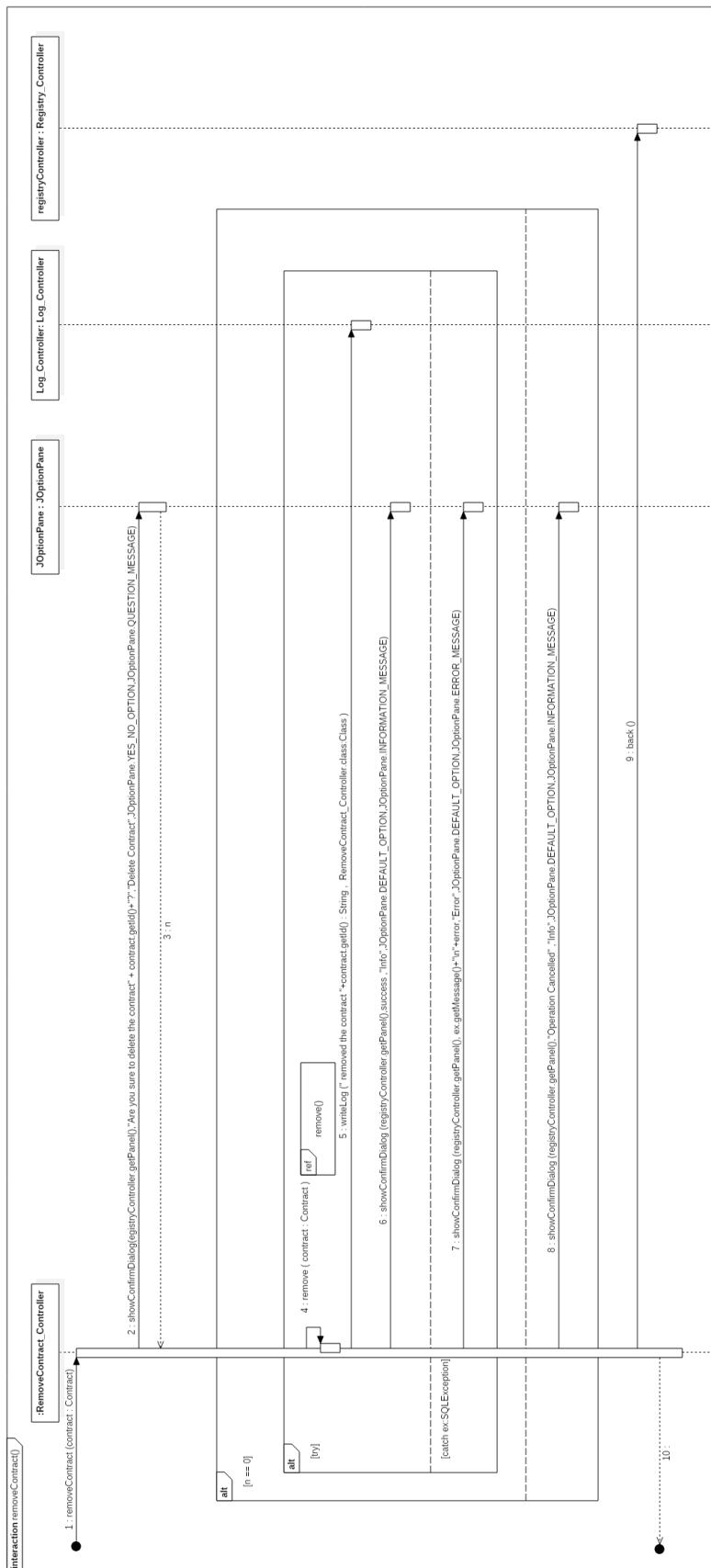


AlterContract_Controller.updateBillingAddressClicked()

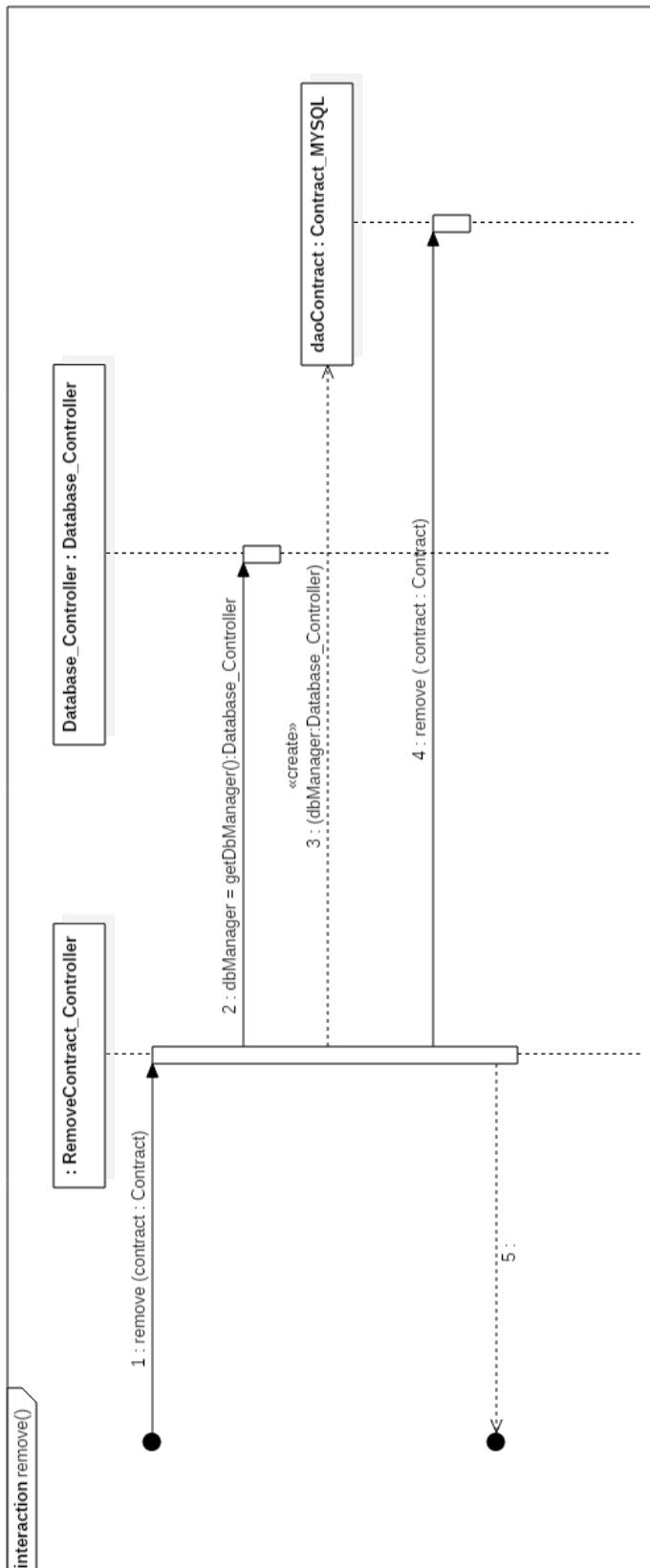
AlterContract_Controller.updateRegistryClicked()



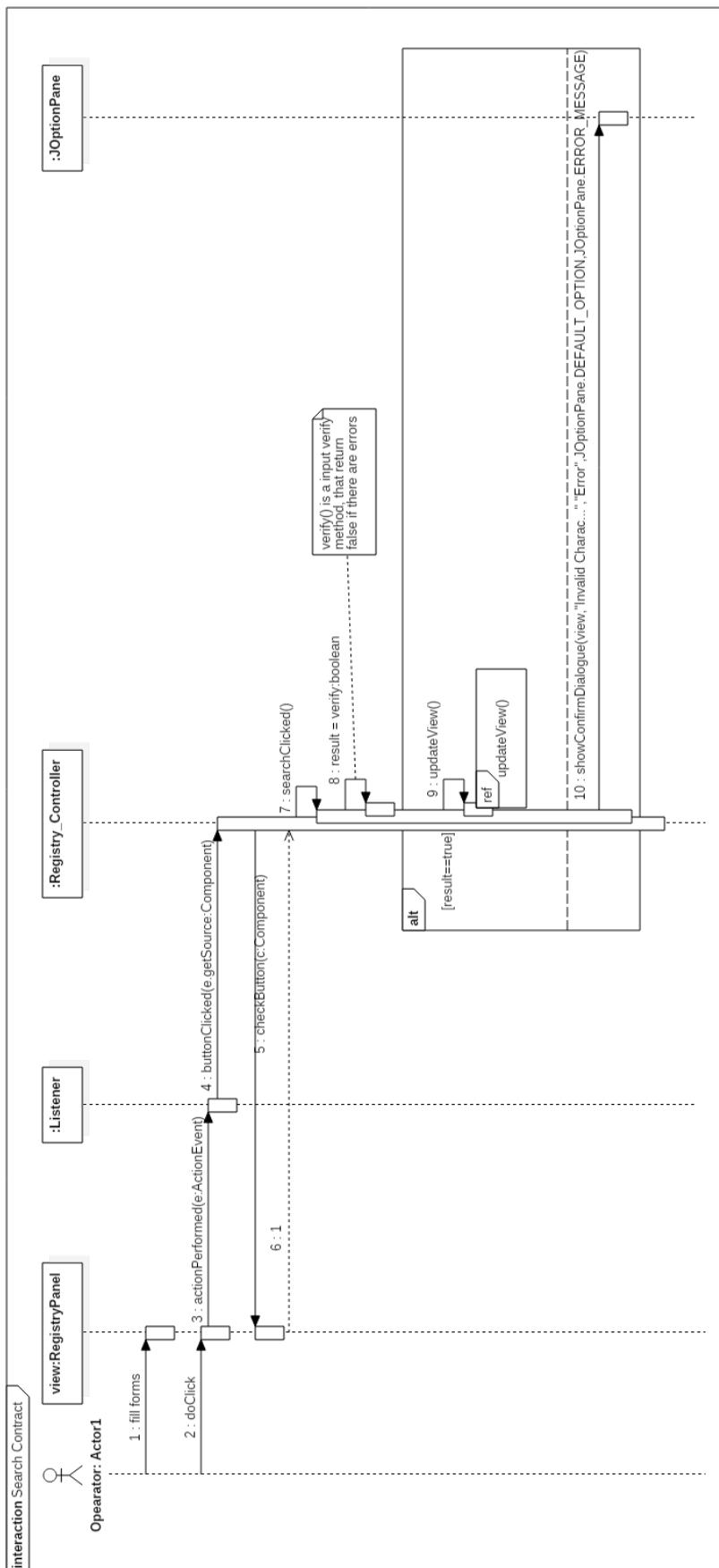
Remove contract

RemoveContract_Controller.removeContract()

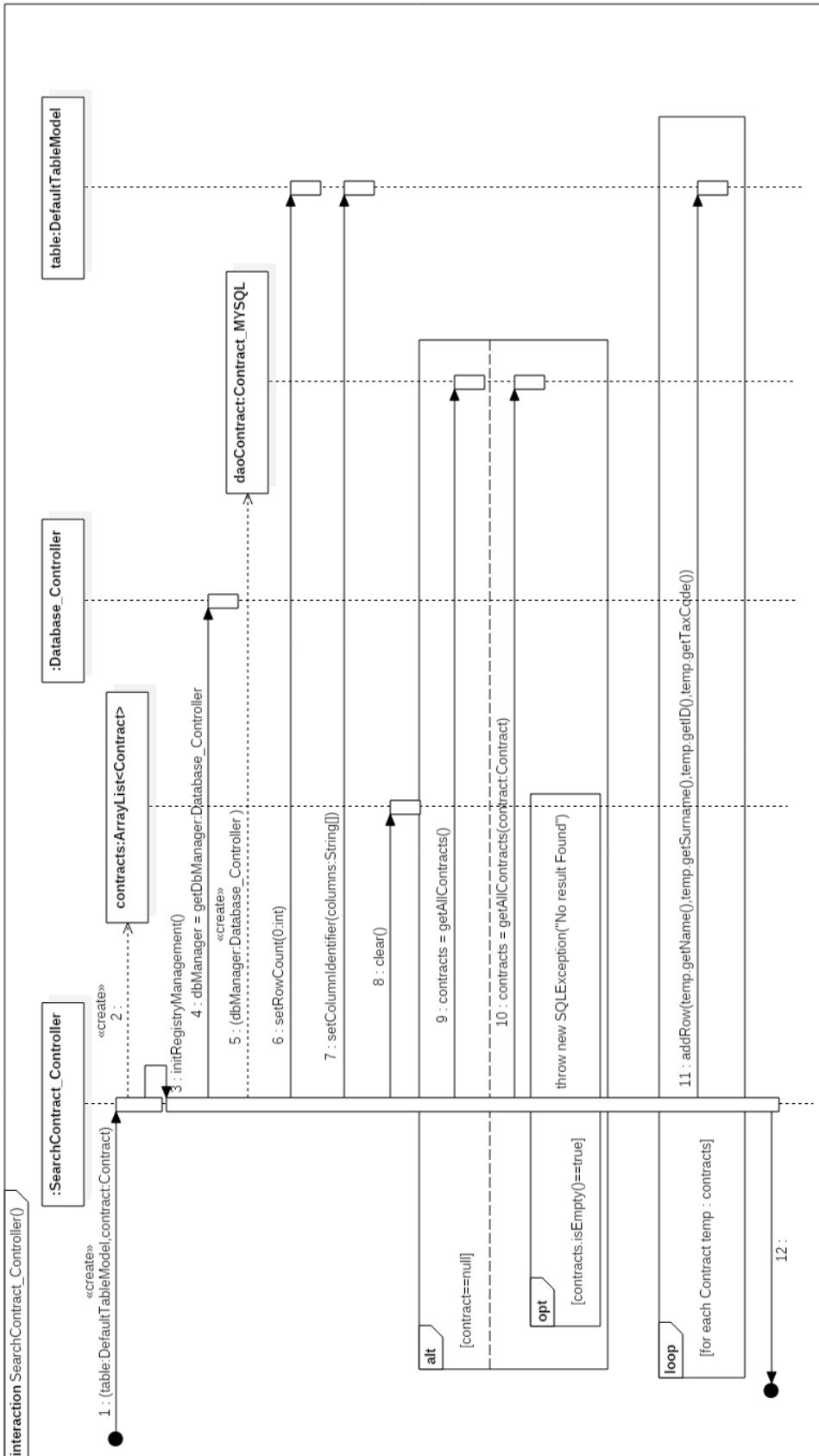
RemoveContract_Controller().remove()

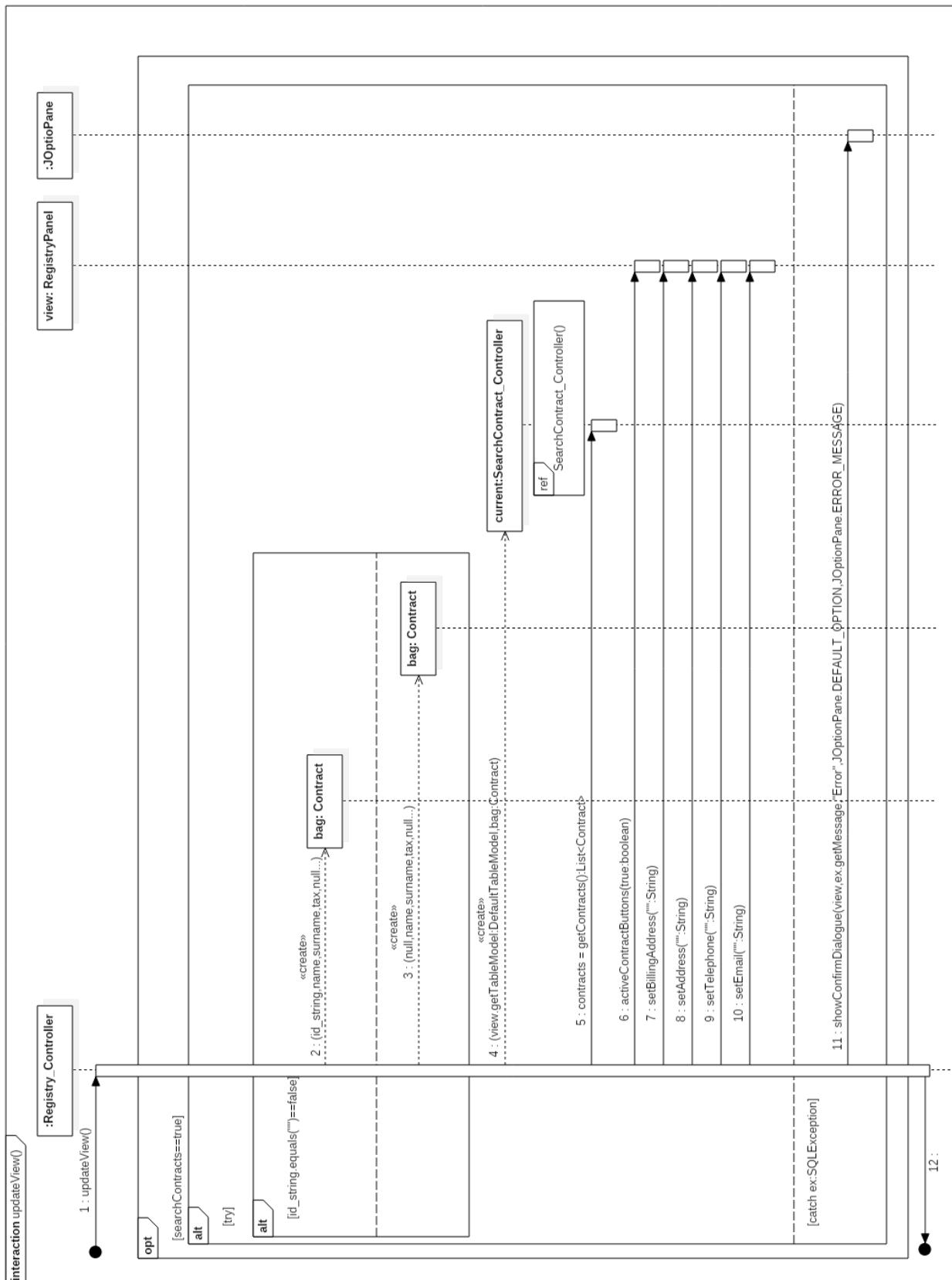


Search contract

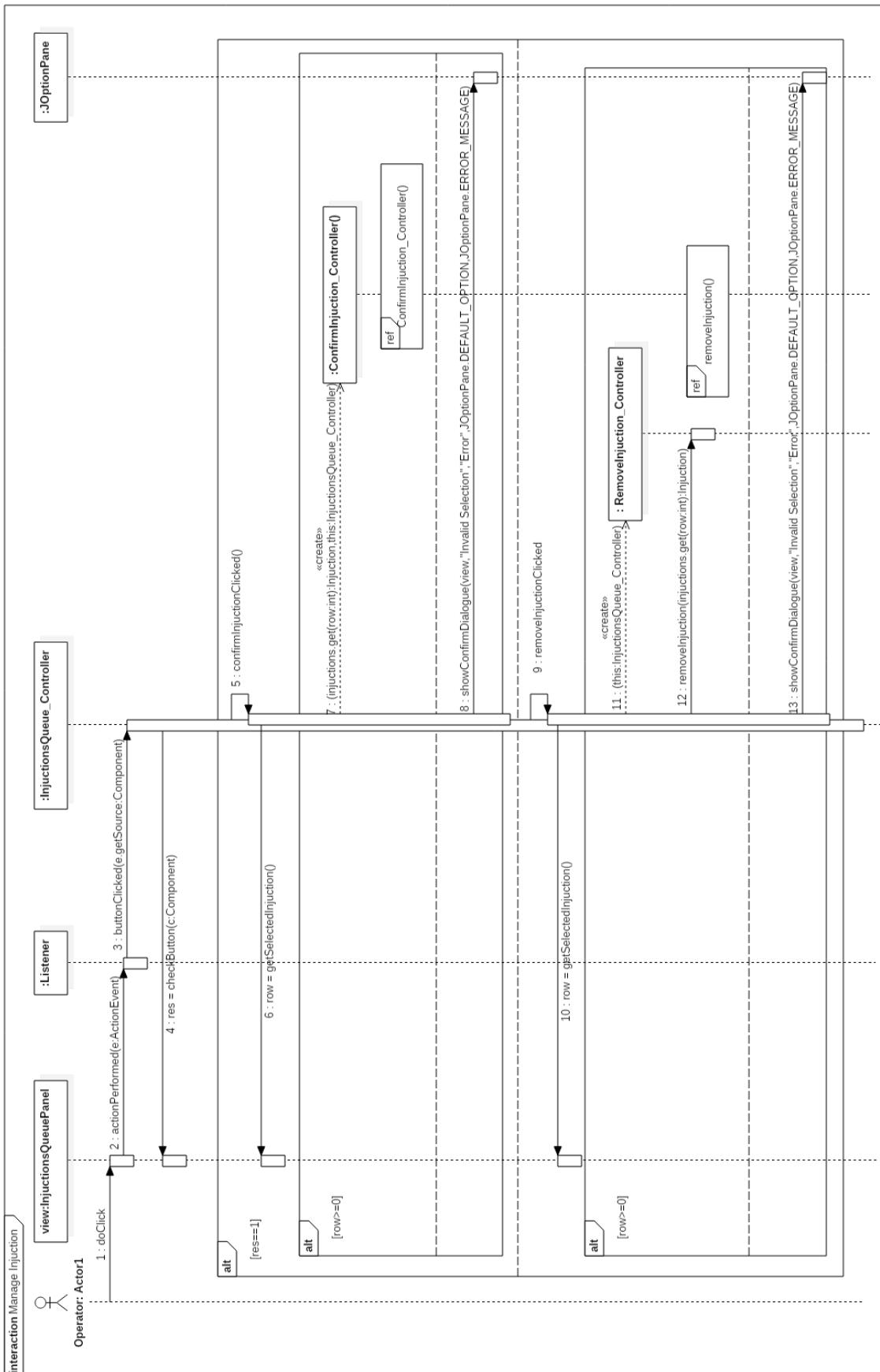


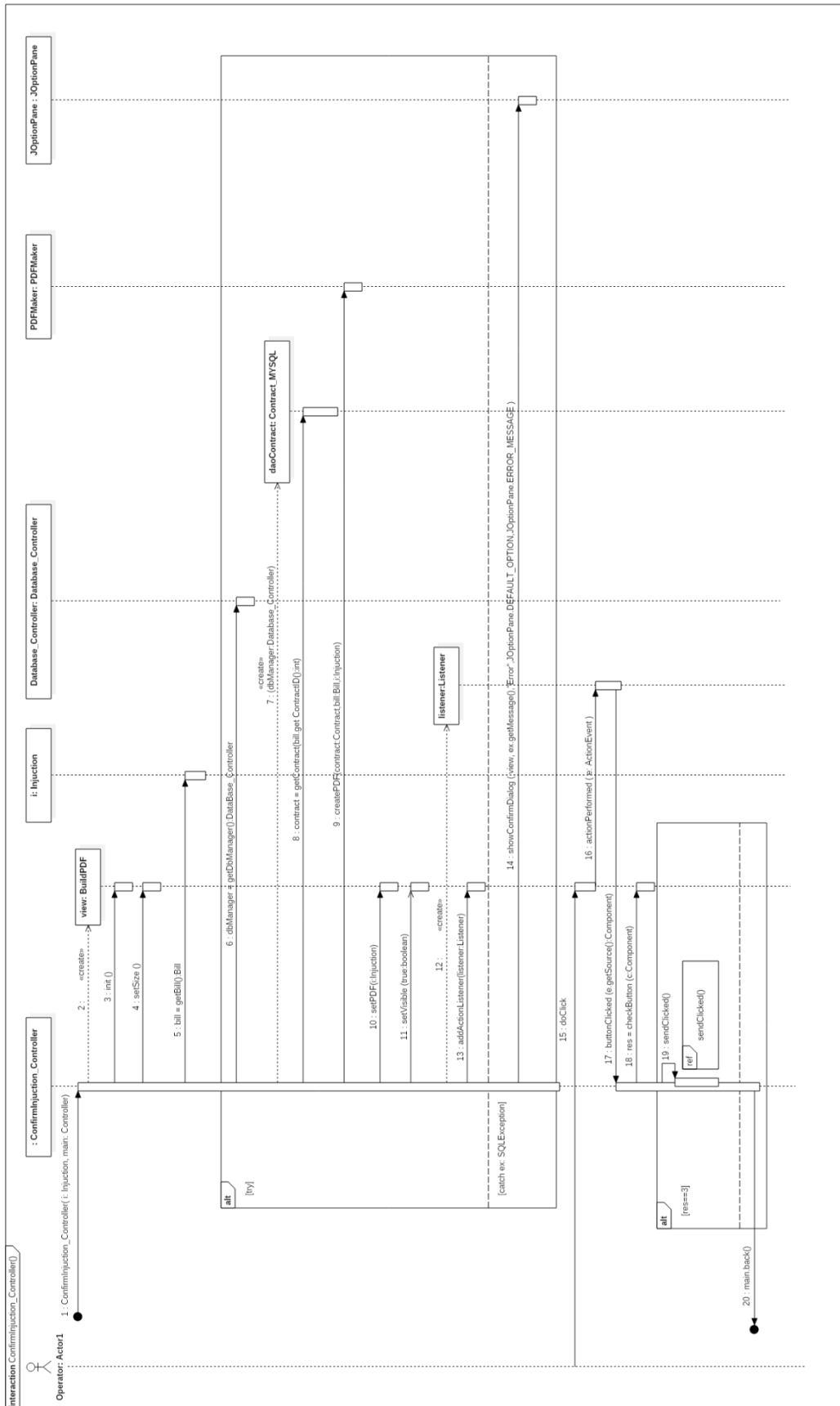
SearchContract_Controller()

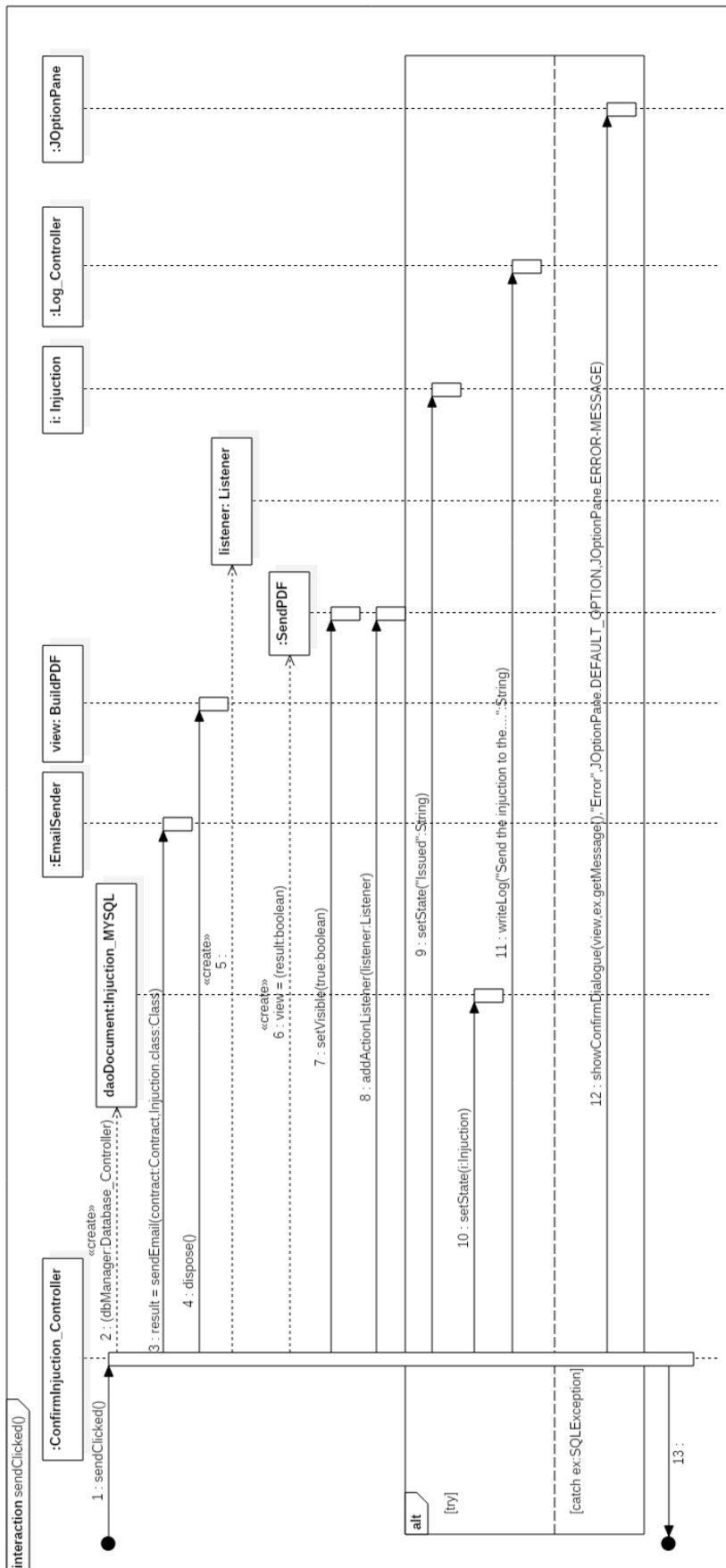


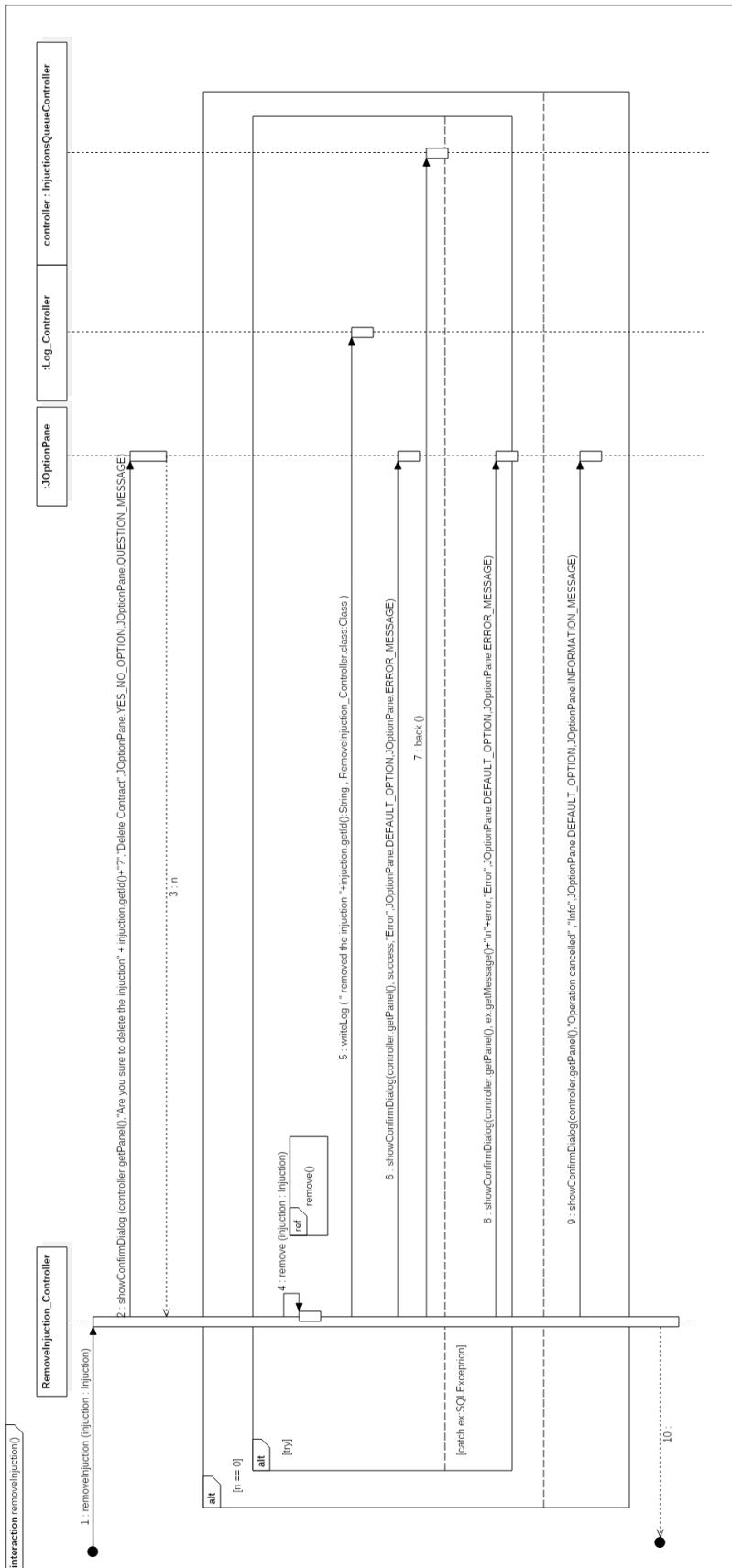
SearchContract_Controller.updateView()

Manage injunction

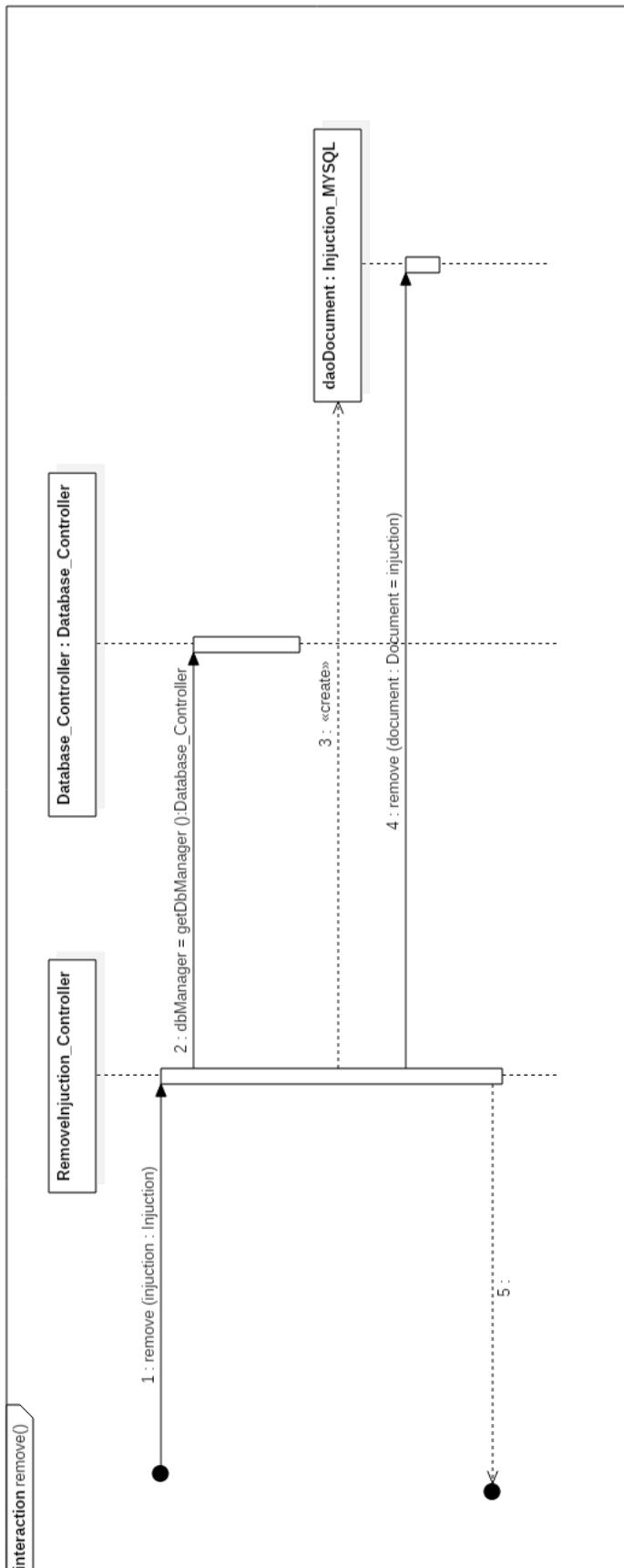


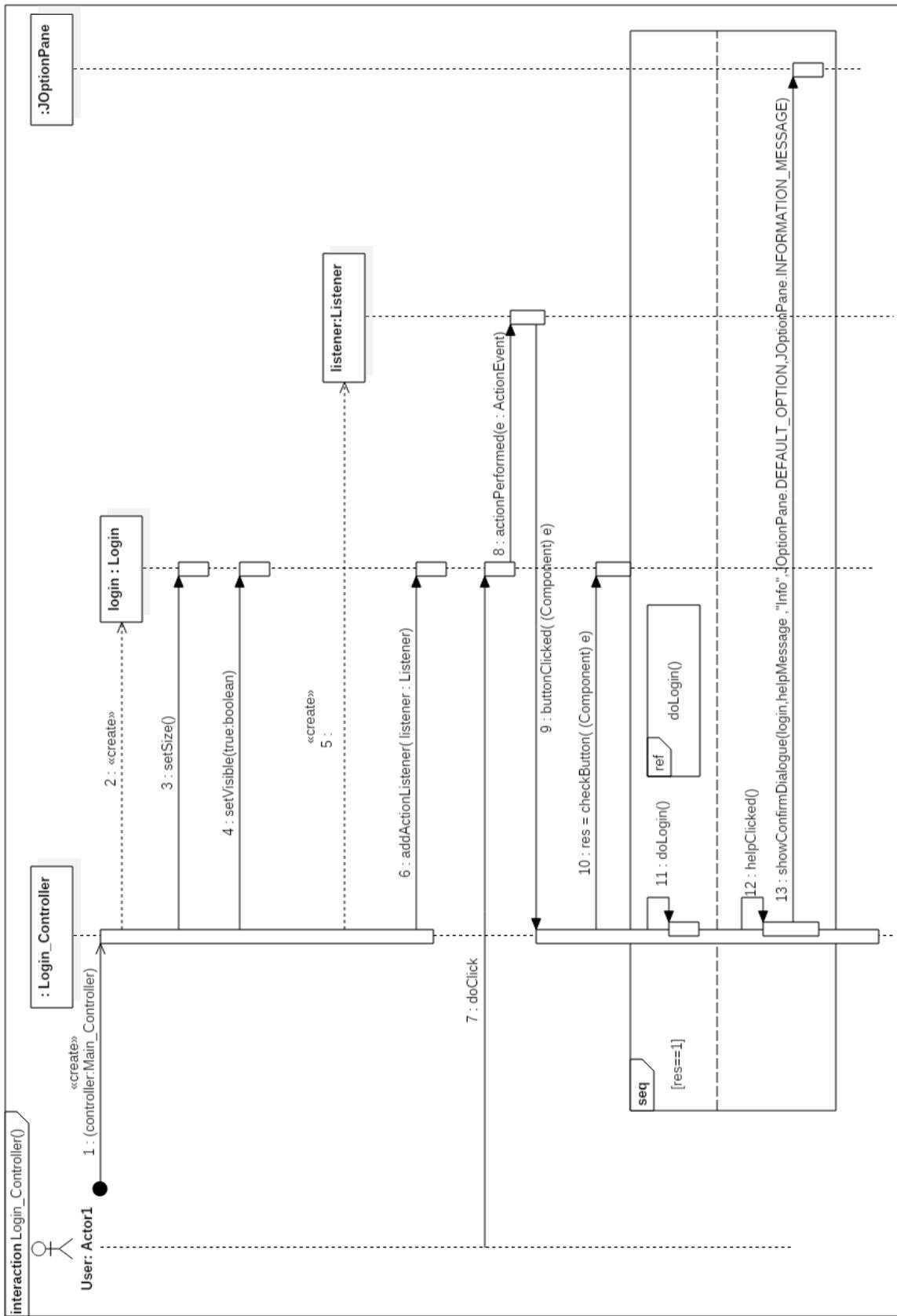
ConfirmInjunction_Controller()

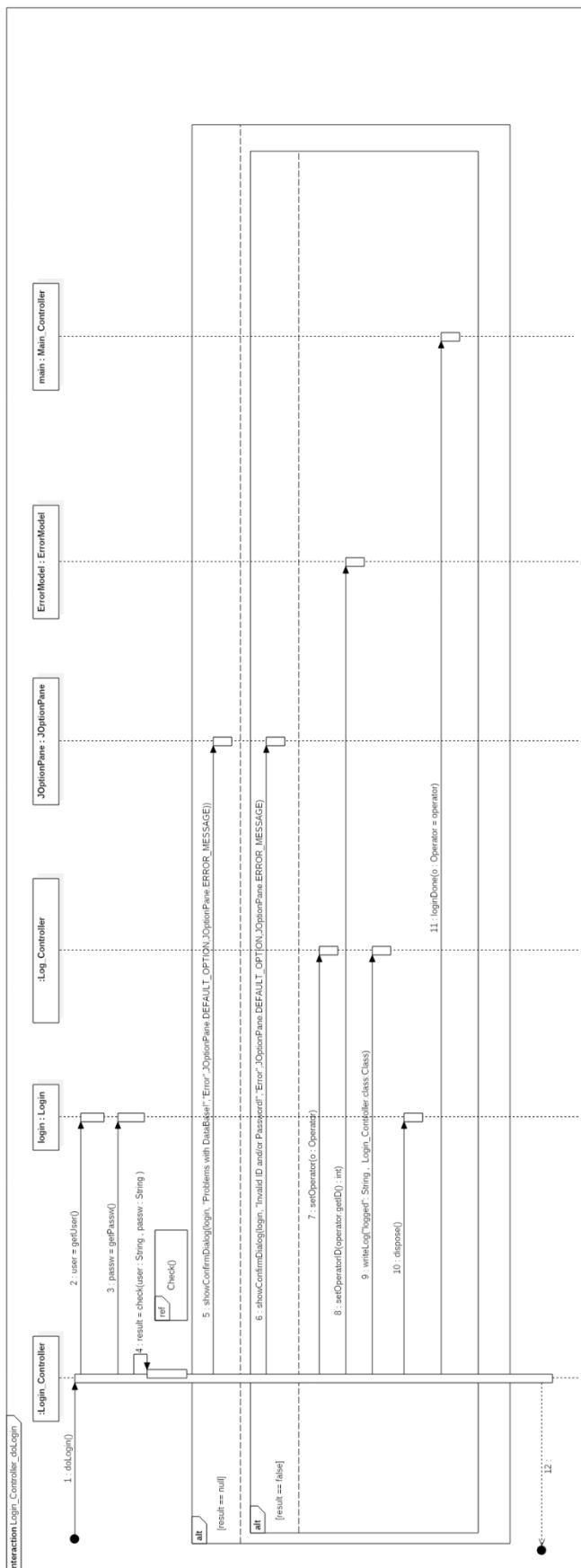
ConfirmInjuntion_Controller.sendClicked()

RemoveInjunction_Controller.removeInjunction()

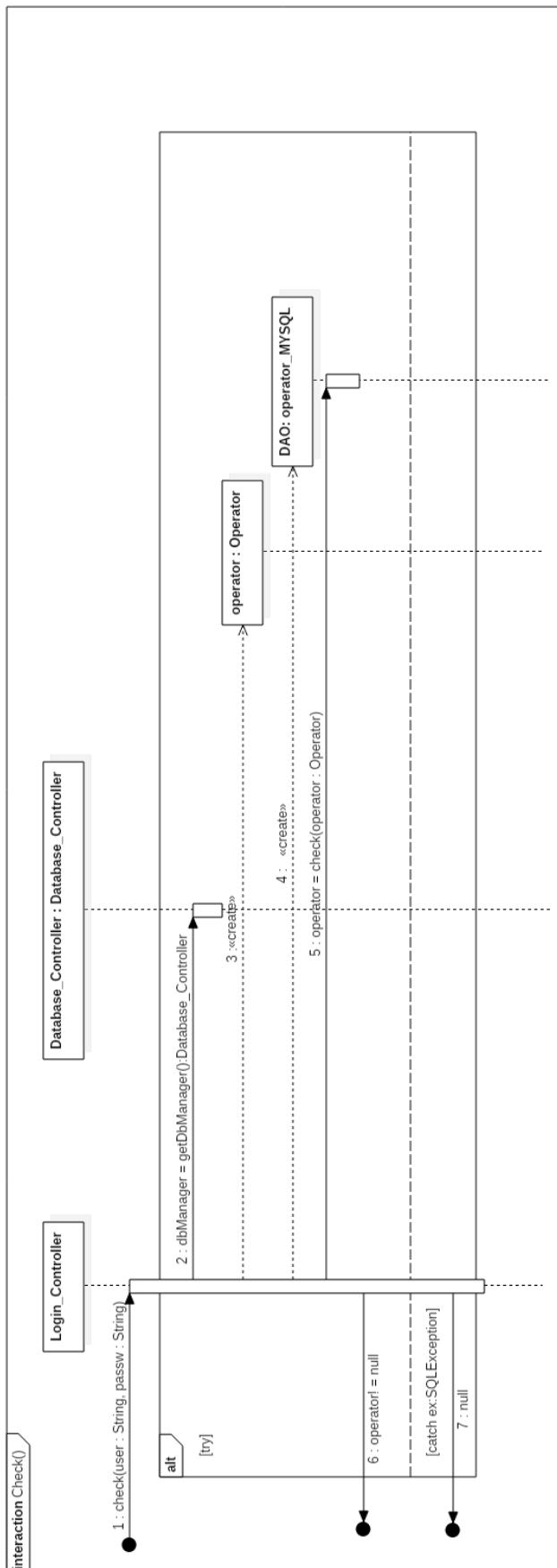
RemoveInjunction_Controller.remove()



Login_Controller()

Login_Controller.doLogin()

Login_Controller.check()

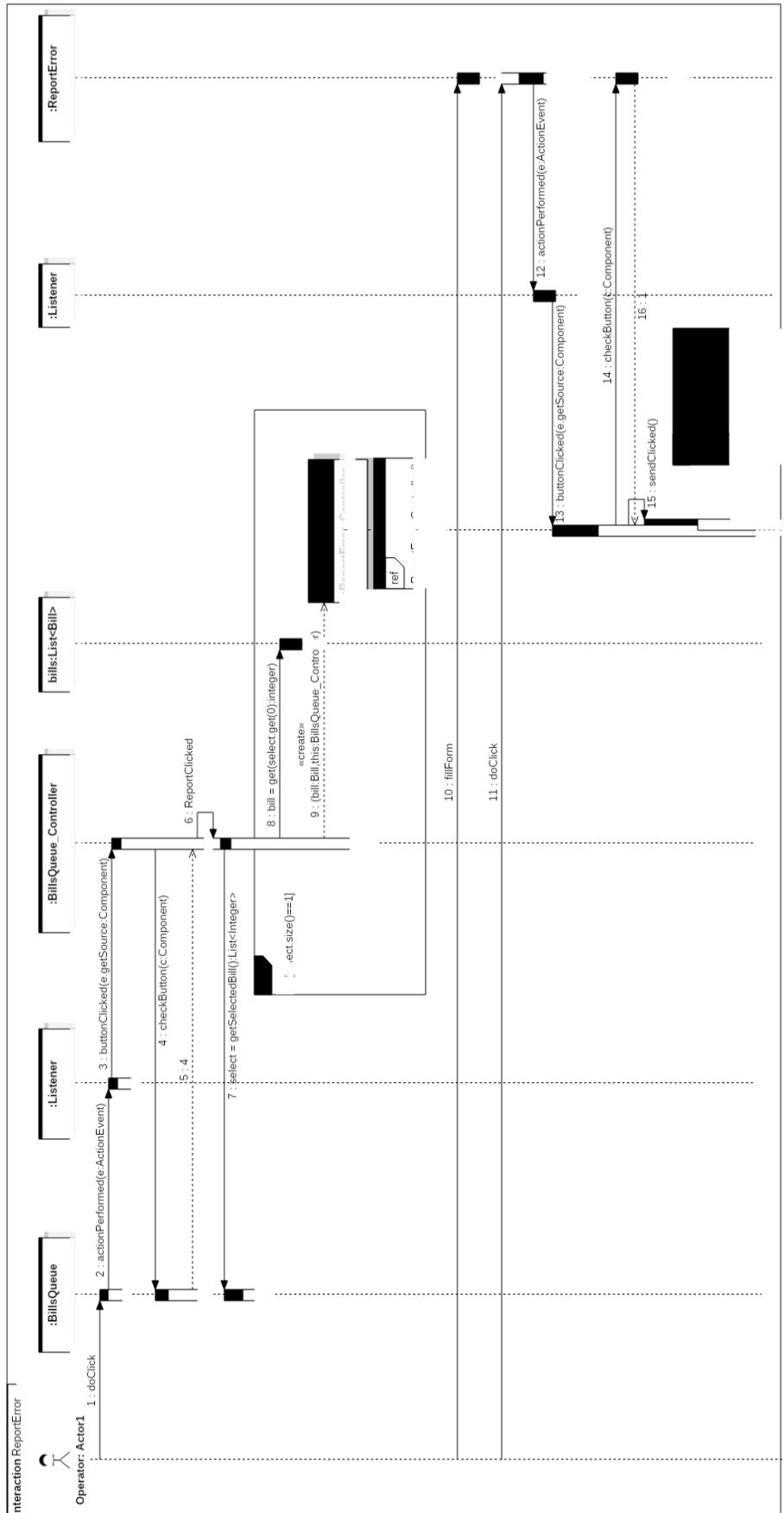


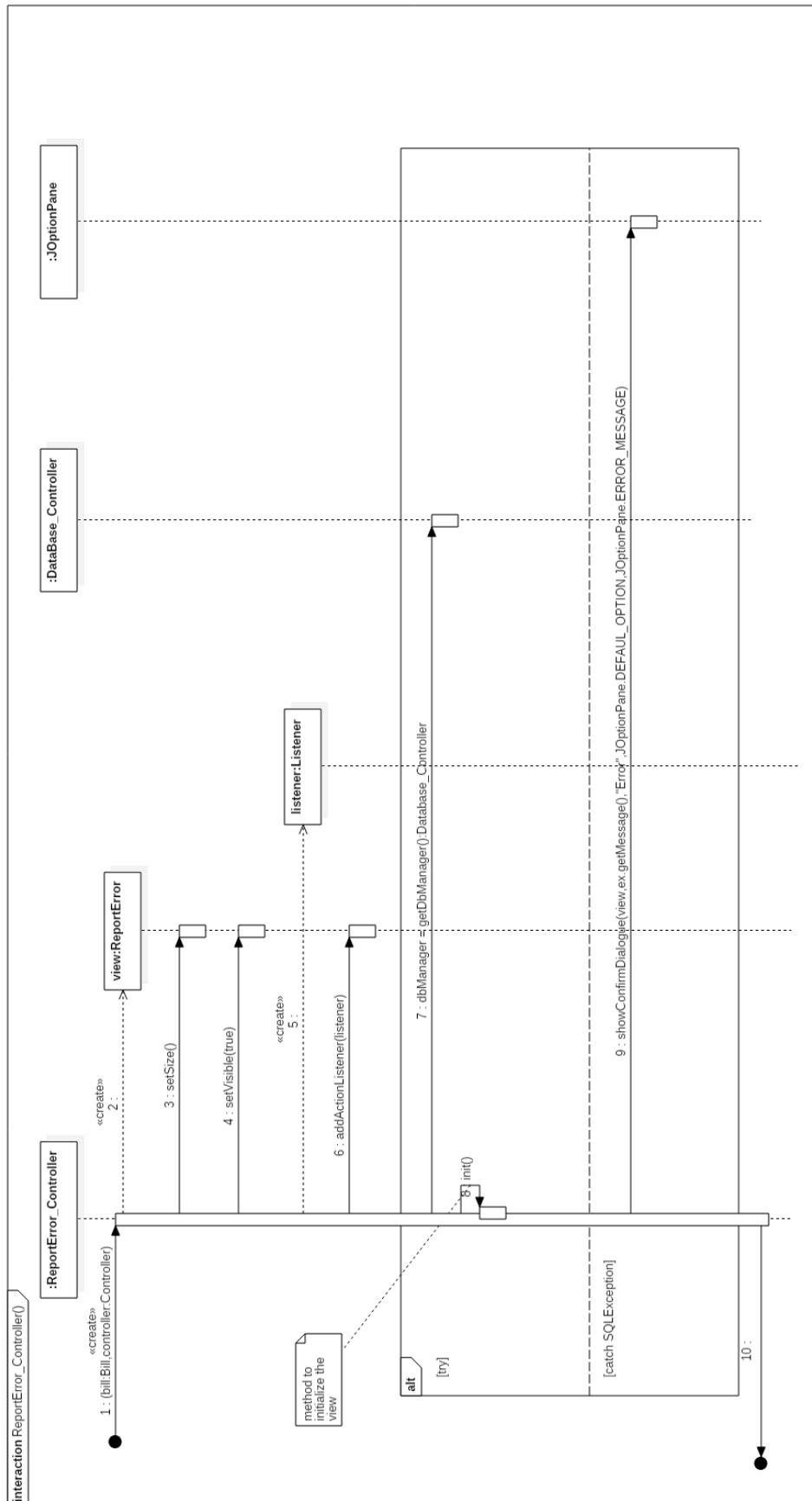
System design document

GCI16

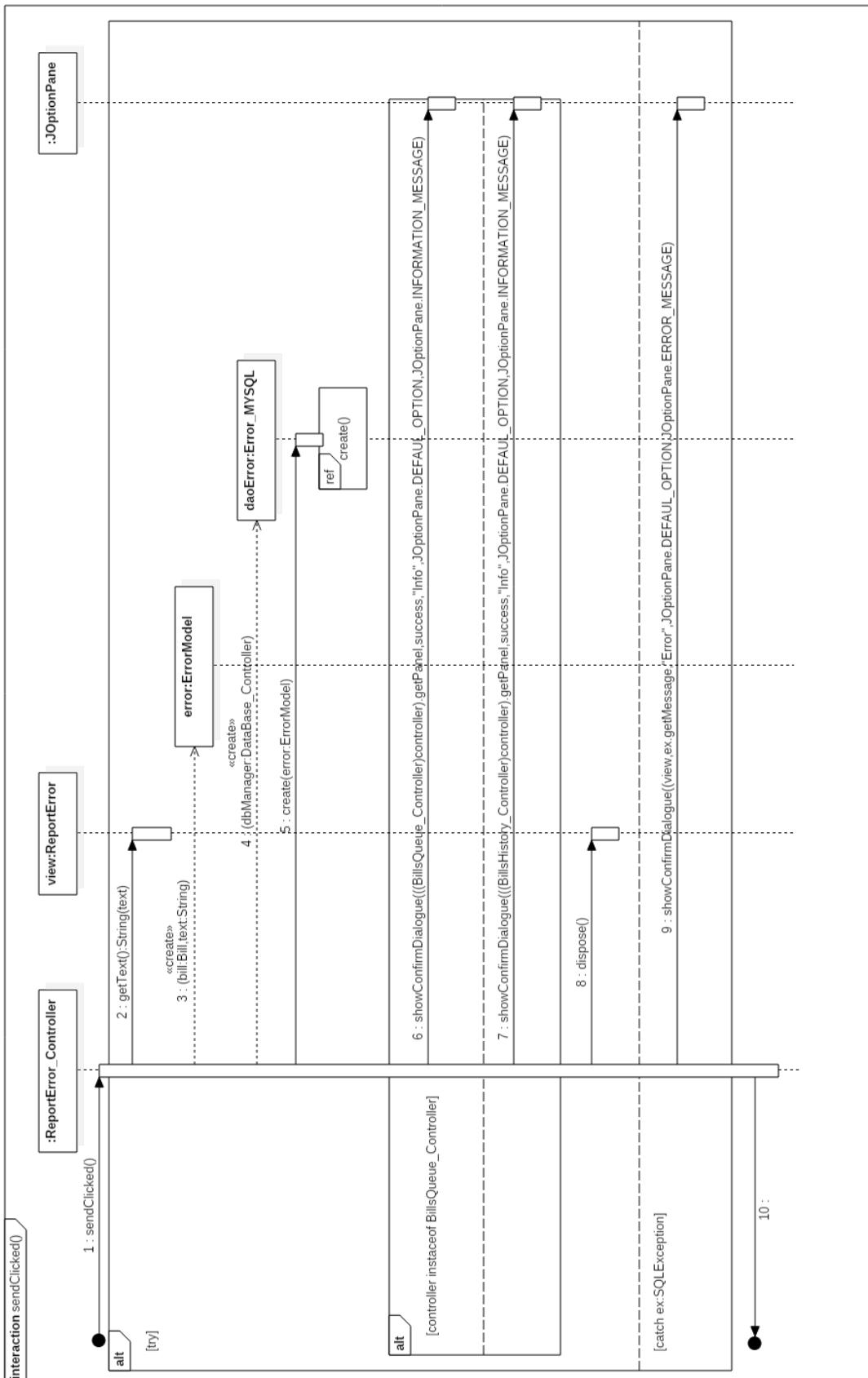


Report error

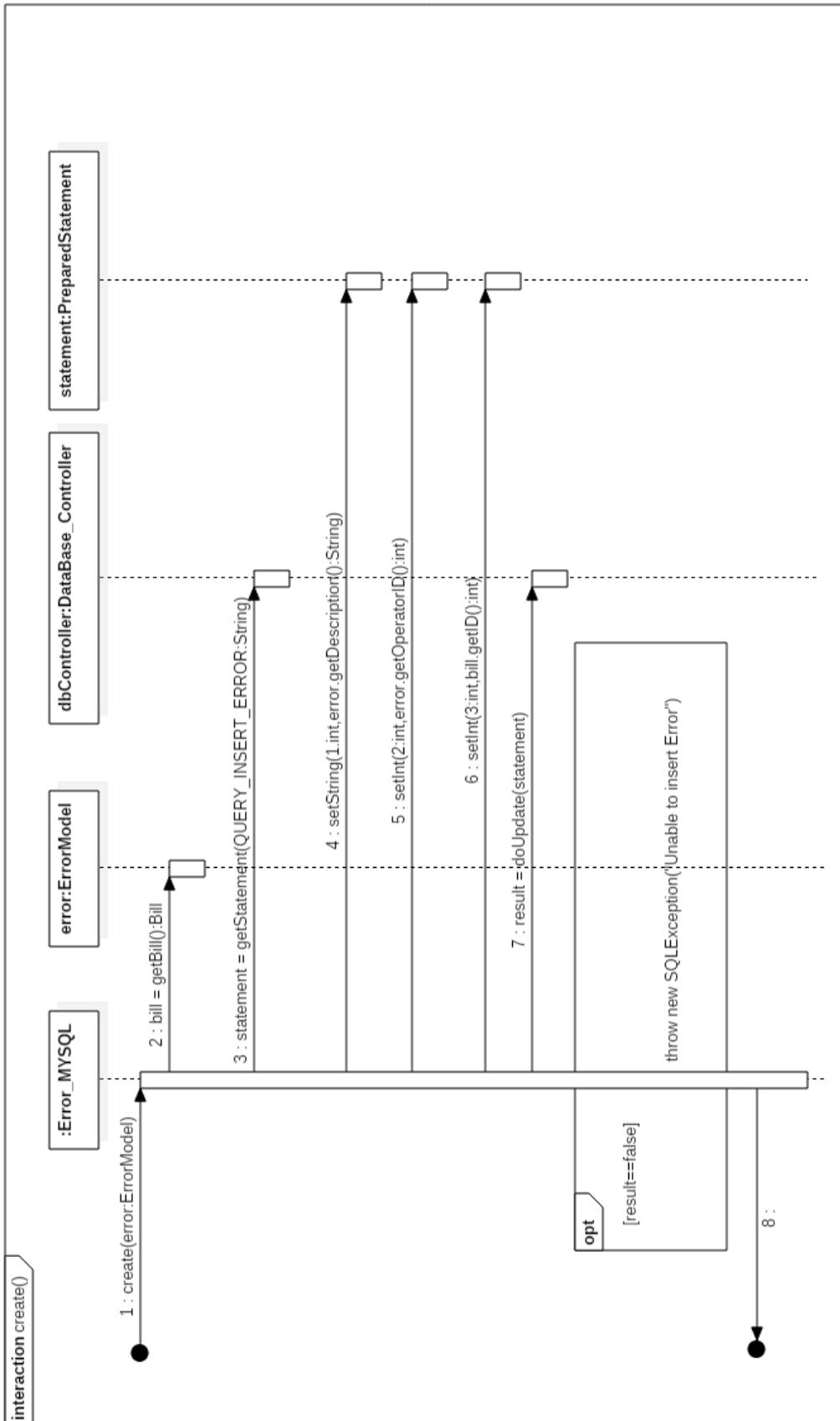


ReportError_Controller()

ReportError_Controller.sendClicked()

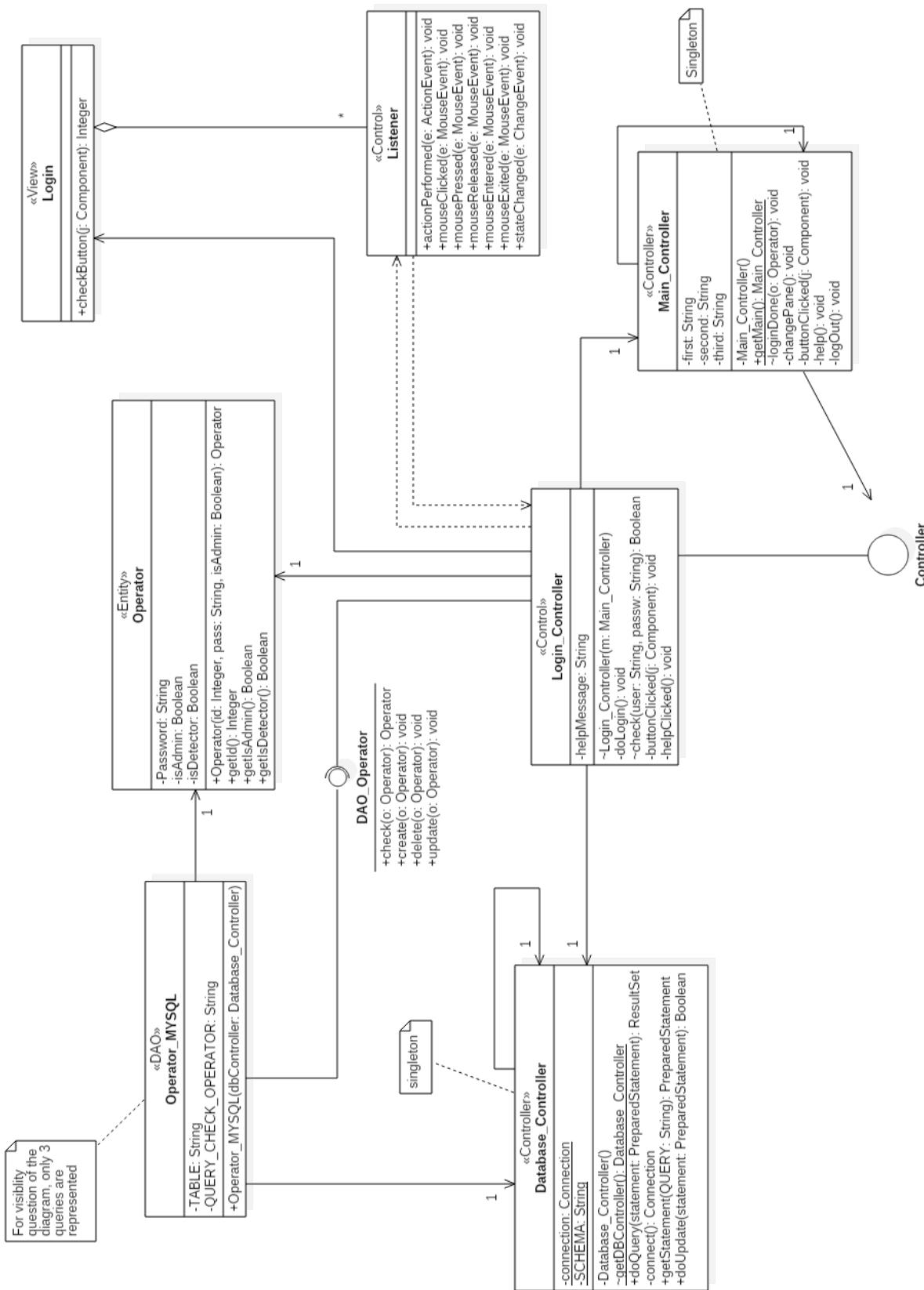


Error_MYSQL.create()

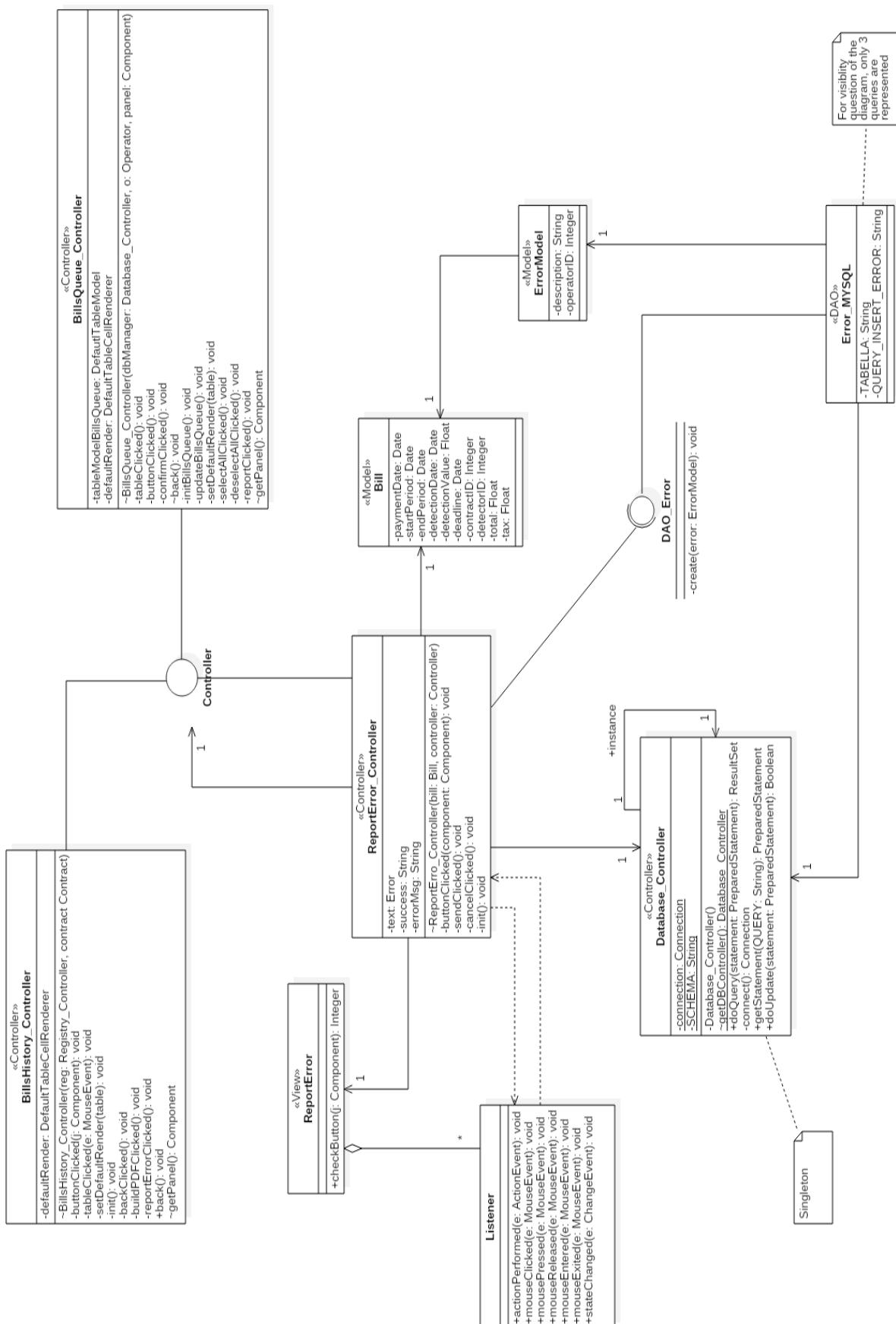


Design class diagram

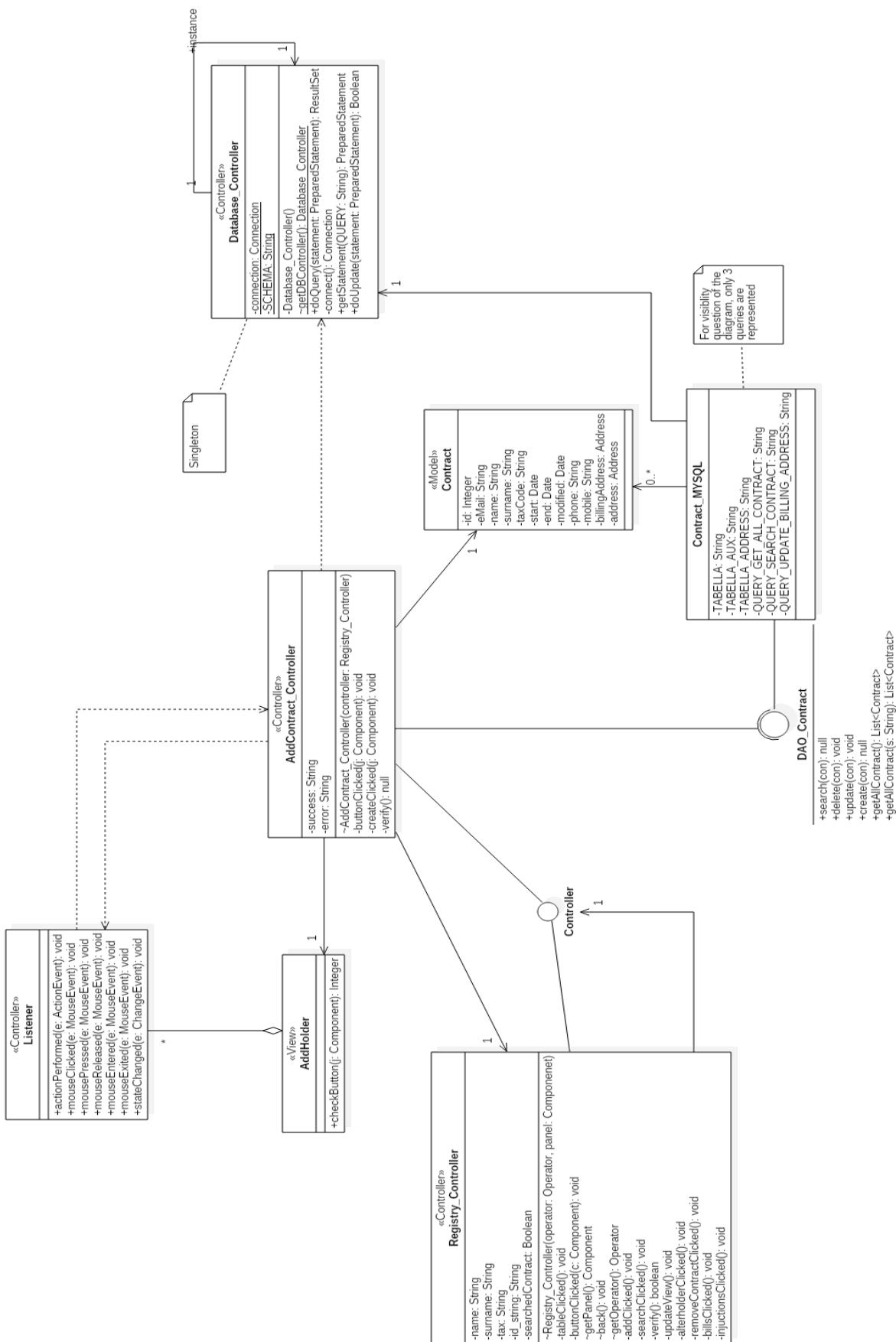
Performs login



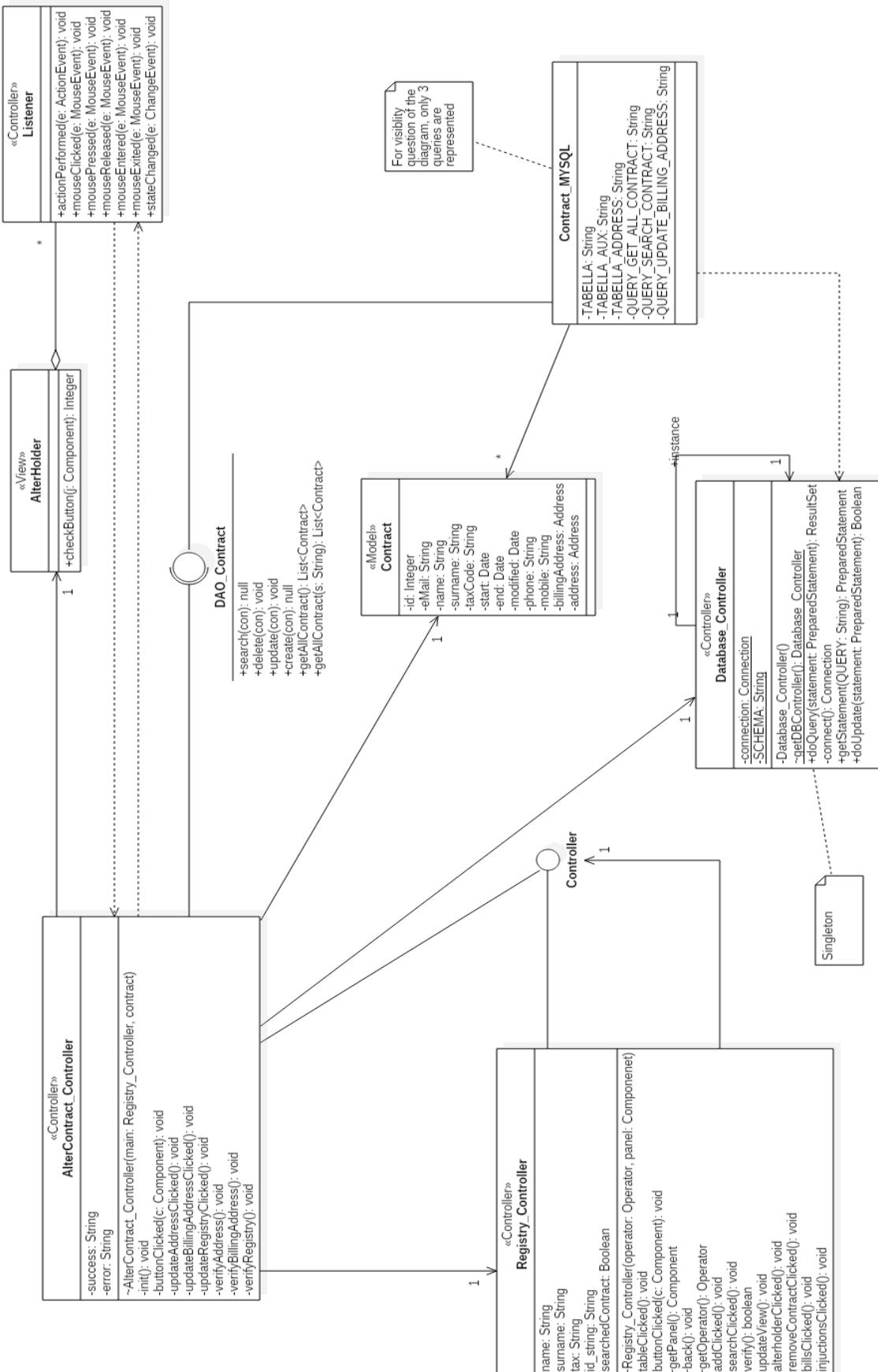
Report error in bill



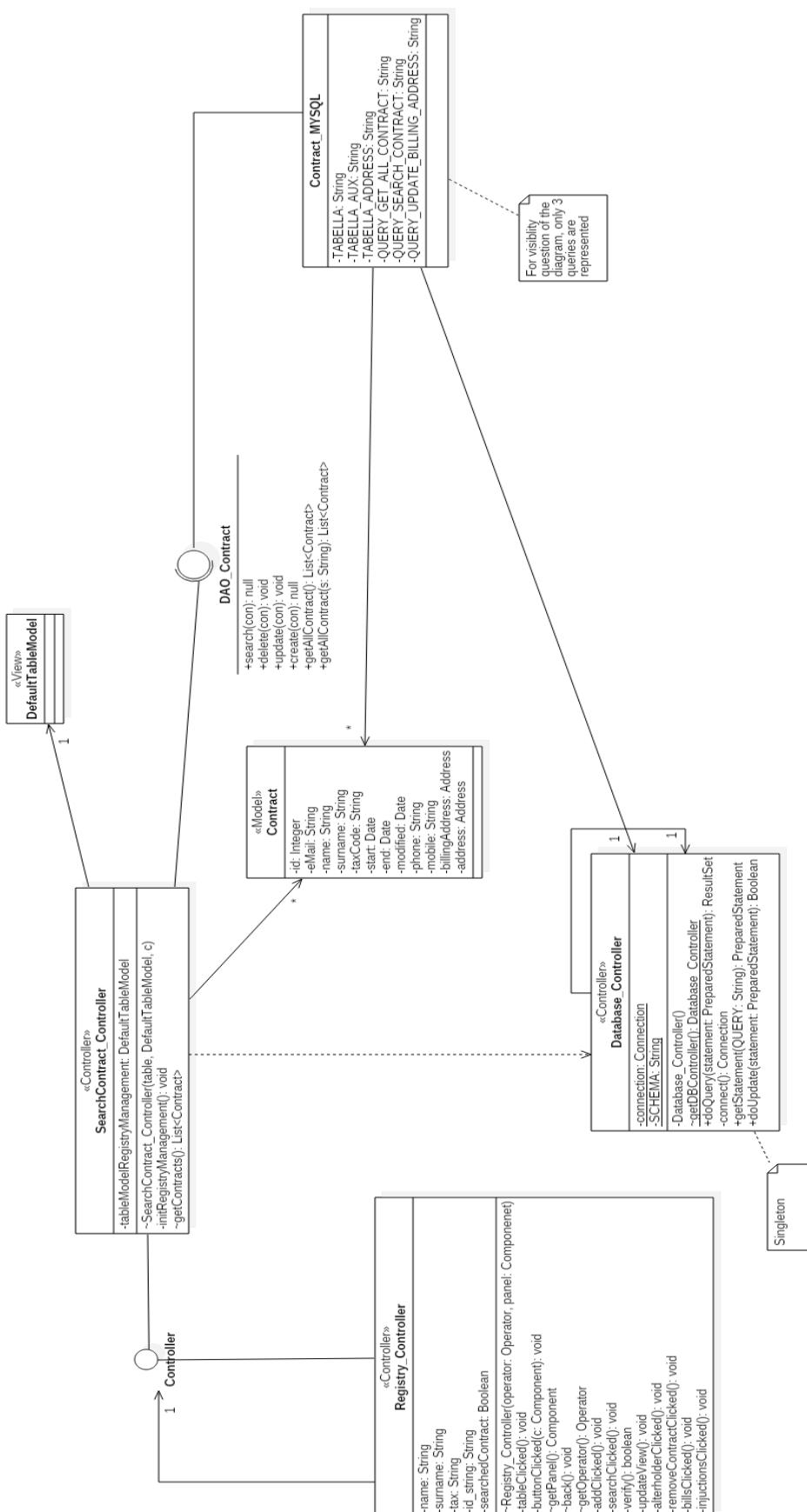
Add contract



Alter contract



Search contract



System design document

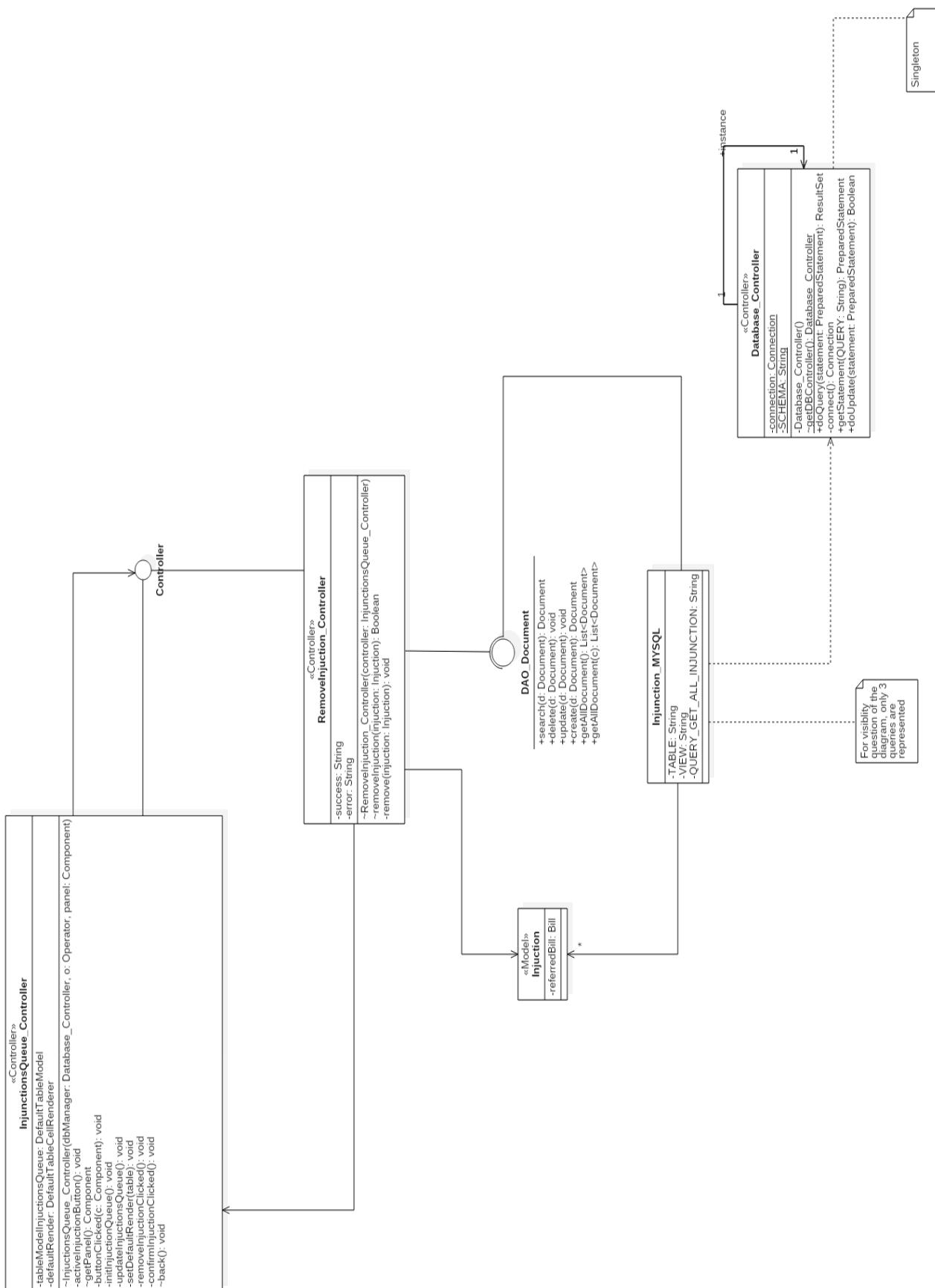
GCI16

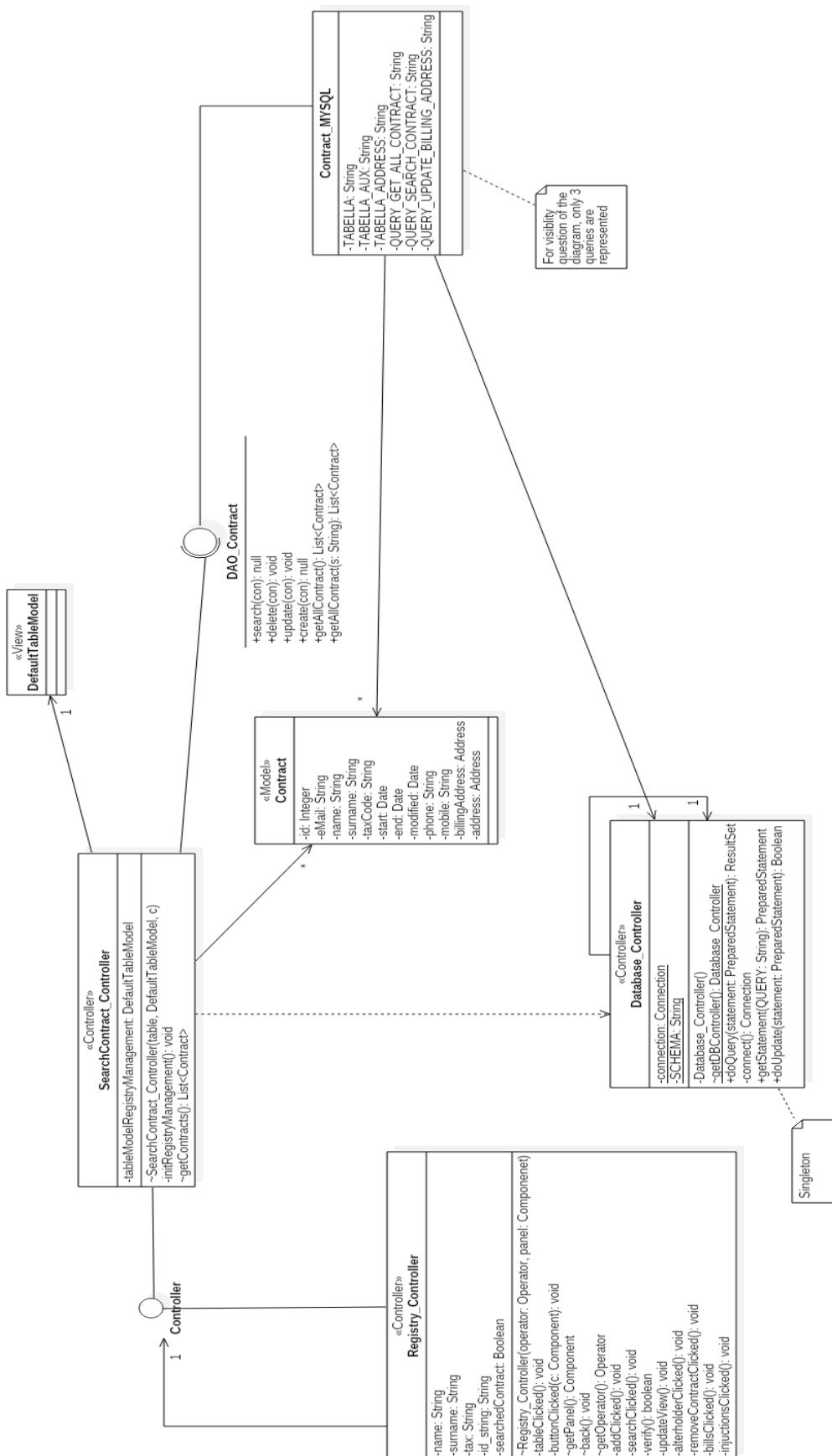


Confirm injunction



Delete injunction



Listener

Testing document

System testing

Performs login test

| | |
|------------------|--|
| Test ID | 1 |
| Test name | Performs login |
| Test goal | The goal is to let the operator log into the system inserting the right credentials |
| Note | In this case we consider these credentials: "User1", "Psw1": right credentials (stored into the DBMS); "User2", "Psw2": wrong credentials (non-stored into the DBMS) |

| Input | Desired output | Obtained output |
|---|---|---|
| The operator fills the form with credentials: "User1", "Psw1" | System shows the "Registry Management" view | System shows the "Registry Management" view |
| The operator fills the form with credentials: "User1", "Psw2" | System shows the error's popup with message "Invalid ID and/or Password!" | System shows the error's popup with message "Invalid ID and/or Password!" |
| The operator fills the form with credentials: "User2", "Psw1" | System shows the error's popup with message "Invalid ID and/or Password!" | System shows the error's popup with message "Invalid ID and/or Password!" |
| The operator fills the form with credentials: "User2", "Psw2" | System shows the error's popup with message "Invalid ID and/or Password!" | System shows the error's popup with message "Invalid ID and/or Password!" |

Researches contact test

| | |
|------------------|---|
| Test ID | 2 |
| Test name | Researches contract |
| Test goal | The goal is to let the operator research contracts using forms |
| Note | In this case we consider these credentials: "Name1", "Surname1", "ContractID1", "TaxC/VAT1": right credentials (stored into the DBMS); "Name2", "Surname2", "ContractID2", "TaxC/VAT2": wrong credentials (non-stored into the DBMS); |

| Input | Desired output | Obtained output |
|---|---|---|
| The operator fills the form with credentials: "Name1", "Surname1", "ContractID1", "TaxC/VAT1" | System shows the results into the table | System shows the results into the table |
| The operator fills the form with credentials: "Name2" | System shows the error's popup with message "No result found" | System shows the error's popup with message "No result found" |
| The operator fills the form with credentials: "Surname2" | System shows the error's popup with message "No result found" | System shows the error's popup with message "No result found" |
| The operator fills the form with credentials: "ContractID2" | System shows the error's popup with message "No result found" | System shows the error's popup with message "No result found" |
| The operator fills the form with credentials: "TaxC/VAT2" | System shows the error's popup with message "No result found" | System shows the error's popup with message "No result found" |

Alters contract test

| | |
|------------------|--|
| Test ID | 3 |
| Test name | Alters contract |
| Test goal | The goal is to let the operator alter a contract selected from the table |
| Note | |

| Input | Desired output | Obtained output |
|---|--|--|
| The operator fills the form with new credentials | System shows a popup with message “Contract altered” displayed | System shows a popup with message “Contract altered” displayed |
| The operator fills the form with invalid characters | System shows a popup with message “Invalid characters” | System shows a popup with message “Invalid characters” |

Deletes contract test

| | |
|------------------|---|
| Test ID | 4 |
| Test name | Deletes contract |
| Test goal | The goal is to let the operator delete a contract selected from the table |
| Note | We consider three cases: "Contract1": this contract can be removed; "Contract2": this contract can't be removed because it has some pendant bills; "Contract3": this contract can't be removed because is already closed. |

| Input | Desired output | Obtained output |
|---------------------------------------|---|---|
| The operator selected the "Contract1" | System shows a popup with message "Contract removed" | System shows a popup with message "Contract removed" |
| The operator selected the "Contract2" | System shows the error's popup with a message that explains the reason why it can't be closed | System shows the error's popup with a message that explains the reason why it can't be closed |
| The operator selected the "Contract3" | System shows the error's popup with a message that explains the reason why it can't be closed | System shows the error's popup with a message that explains the reason why it can't be closed |

Deletes contract test

| | |
|------------------|---|
| Test ID | 4 |
| Test name | Deletes contract |
| Test goal | The goal is to let the operator delete a contract selected from the table |
| Note | We consider three cases: "Contract1": this contract can be removed; "Contract2": this contract can't be removed because it has some pendant bills; "Contract3": this contract can't be removed because is already closed. |

| Input | Desired output | Obtained output |
|---------------------------------------|---|---|
| The operator selected the "Contract1" | System shows a popup with message "Contract removed" | System shows a popup with message "Contract removed" |
| The operator selected the "Contract2" | System shows the error's popup with a message that explains the reason why it can't be closed | System shows the error's popup with a message that explains the reason why it can't be closed |
| The operator selected the "Contract3" | System shows the error's popup with a message that explains the reason why it can't be closed | System shows the error's popup with a message that explains the reason why it can't be closed |

Adds contract test

| | |
|------------------|---|
| Test ID | 5 |
| Test name | Adds contract |
| Test goal | The goal is to let the operator add new contracts |
| Note | |

| Input | Desired output | Obtained output |
|--|--|--|
| The operator fills all forms interested in the contract's adding | System shows a popup with message "New contract added" | System shows a popup with message "New contract added" |
| The operator fills a form with invalid characters | System shows the error's popup with a message that explains the reason why it can't continue the operation | System shows the error's popup with a message that explains the reason why it can't continue the operation |

Reports errors in bills test

| | |
|------------------|---|
| Test ID | 6 |
| Test name | Reports errors in bills |
| Test goal | The goal is to let the operator reports errors in bills |
| Note | |

| Input | Desired output | Obtained output |
|---|--|--|
| The operator fills the form with the errors | System shows a popup with message "Operation successfully completed" | System shows a popup with message "Operation successfully completed" |

Deletes injunctions test

| | |
|------------------|--|
| Test ID | 7 |
| Test name | Deletes injunctions |
| Test goal | The goal is to let the operator delete injunctions |
| Note | |

| Input | Desired output | Obtained output |
|---|--|--|
| The operator clicked the "Remove" button | System shows a popup with message "Operation successfully completed" | System shows a popup with message "Operation successfully completed" |
| The operator clicked the "Remove" button but the injunction can't be removed because it has pending payment | System shows a popup with an error message | System shows a popup with an error message |

Confirms bill test

| | |
|------------------|---|
| Test ID | 8 |
| Test name | Confirms bill |
| Test goal | The goal is to let the operator confirm a selected bill |
| Note | |

| Input | Desired output | Obtained output |
|--|---------------------------------------|---------------------------------------|
| The operator selects a bill from the bill's table and presses "Confirm button" | System shows the bill's preview (PDF) | System shows the bill's preview (PDF) |

Resends bill test

| | |
|------------------|--|
| Test ID | 9 |
| Test name | Resends bill |
| Test goal | The goal is to let the operator resend a selected bill |
| Note | |

| Input | Desired output | Obtained output |
|--|---|---|
| The operator selects a bill from the bills' table and presses "Confirm button" | System shows the bill's preview (PDF) | System shows the bill's preview (PDF) |
| The operator selects a bill from the bills' table that has been payed and presses "Confirm button" | System shows a popup that contains an error message | System shows a popup that contains an error message |

Confirms injunctions test

| | |
|------------------|---|
| Test ID | 10 |
| Test name | Confirms injunctions |
| Test goal | The goal is to let the operator confirm an injunction |
| Note | |

| Input | Desired output | Obtained output |
|---|---------------------------------------|---------------------------------------|
| The operator selects an injunction from the injunction's table and presses "Confirm button" | System shows the bill's preview (PDF) | System shows the bill's preview (PDF) |

Resends injunctions test

| | |
|------------------|---|
| Test ID | 11 |
| Test name | Resends injunctions |
| Test goal | The goal is to let the operator resends an injunction |
| Note | |

| Input | Desired output | Obtained output |
|---|---|---|
| The operator selects an injunction from the injunction's table and presses "Confirm button" | System shows the bill's preview (PDF) | System shows the bill's preview (PDF) |
| The operator selects an injunction from the injunctions' table that has been payed and presses "Confirm button" | System shows a popup that contains an error message | System shows a popup that contains an error message |
| The operator selects an injunction that refers to a payed bill | System shows a popup that contains an error message | System shows a popup that contains an error message |

JUnit code

Source

```
1. package Controller;
2.
3. import java.sql.SQLException;
4. import java.util.logging.Level;
5. import java.util.logging.Logger;
6. import org.junit.Before;
7. import org.junit.BeforeClass;
8. import org.junit.Test;
9. import static org.junit.Assert.*;
10. /**
11. *
12. * @author Andrea
13. *
14. * This class tests the "check" method of "Login_Controller" class.
15. * This methods takes 2 parameters in input: USER(String) and PSW(String).
16. *
17. * Equivalence classes:
18. * USER:
19. * - Numerical value between 0 and 100 (T)
20. * - Numerical value lower then 0 (F1)
21. * - Numerical value higher then 100 (F2)
22. * - NULL (F3)
23. * - Alphanumerical value (F4)
24. *
25. * PSW:
26. * - Stored values (T)
27. * - Not stored values (F1)
28. * - NULL (F2)
29. *
30. * Method applied: SECT
31. * Number of tests: 15 (5x3)
32. */
33. public class Login_ControllerTest {
34.     private static Login_Controller instance;
35.     private Boolean result;
36.     private String user;
37.     private String psw;
38.     private Boolean expResult;
39.
40.     public Login_ControllerTest() {
41.     }
42.
43.     @BeforeClass
44.     public static void setUpClass() {
45.         try {
46.             instance = new Login_Controller(Main_Controller.getMain());
47.         } catch (SQLException ex) {
48.             Logger.getLogger(Login_ControllerTest.class.getName()).log(Level.SEVERE, null, ex);
49.             fail("Login_Controller instantiation failed");
50.         }
51.     }
52.
53.     @Before
54.     public void setUp() {
55.         user = null;
56.         psw = null;
57.         result = null;
58.         expResult = null;
59.     }
60.
61.     /*
```

```
62. * EQUIVALENCE CLASSES TESTED:  
63. * User: T  
64. * Psw: T  
65. */  
66. @Test  
67. public void checkTest1(){  
68.     user = "2";  
69.     psw = "ingsw";  
70.     expResult = true;  
71.     result = instance.check(user, psw);  
72.     assertEquals(result, expResult);  
73. }  
74.  
75. /*  
76. * EQUIVALENCE CLASSES TESTED:  
77. * User: T  
78. * Psw: F1  
79. */  
80. @Test  
81. public void checkTest2(){  
82.     user = "2";  
83.     psw = "ingw";  
84.     expResult = false;  
85.     result = instance.check(user, psw);  
86.     assertEquals(result, expResult);  
87. }  
88.  
89. /*  
90. * EQUIVALENCE CLASSES TESTED:  
91. * User: T  
92. * Psw: F2  
93. */  
94. @Test  
95. public void checkTest3(){  
96.     user = "2";  
97.     expResult = false;  
98.     result = instance.check(user, psw);  
99.     assertEquals(result, expResult);  
100. }  
101.  
102. /*  
103. * EQUIVALENCE CLASSES TESTED:  
104. * User: F2  
105. * Psw: T  
106. */  
107. @Test  
108. public void checkTest4(){  
109.     user = "101";  
110.     psw = "ingsw";  
111.     expResult = false;  
112.     result = instance.check(user, psw);  
113.     assertEquals(result, expResult);  
114. }  
115.  
116. /*  
117. * EQUIVALENCE CLASSES TESTED:  
118. * User: F2  
119. * Psw: F1  
120. */  
121. @Test  
122. public void checkTest5(){  
123.     user = "101";  
124.     psw = "ingw";  
125.     expResult = false;  
126.     result = instance.check(user, psw);  
127.     assertEquals(result, expResult);
```

```
128.    }
129.
130.   /*
131.   * EQUIVALENCE CLASSES TESTED:
132.   * User: F2
133.   * Psw: F2
134.   */
135. @Test
136. public void checkTest6(){
137.     user = "101";
138.     expResult = false;
139.     result = instance.check(user, psw);
140.     assertEquals(result, expResult);
141. }
142.
143. /*
144. * EQUIVALENCE CLASSES TESTED:
145. * User: F3
146. * Psw: T
147. */
148. @Test
149. public void checkTest7(){
150.     user = "2abc";
151.     psw = "ingsw";
152.     expResult = false;
153.     result = instance.check(user, psw);
154.     assertEquals(result, expResult);
155. }
156.
157. /*
158. * EQUIVALENCE CLASSES TESTED:
159. * User: F3
160. * Psw: F1
161. */
162. @Test
163. public void checkTest8(){
164.     user = "2abc";
165.     psw = "ingw";
166.     expResult = false;
167.     result = instance.check(user, psw);
168.     assertEquals(result, expResult);
169. }
170.
171. /*
172. * EQUIVALENCE CLASSES TESTED:
173. * User: F3
174. * Psw: F2
175. */
176. @Test
177. public void checkTest9(){
178.     user = "2abc";
179.     expResult = false;
180.     result = instance.check(user, psw);
181.     assertEquals(result, expResult);
182. }
183.
184. /*
185. * EQUIVALENCE CLASSES TESTED:
186. * User: F4
187. * Psw: T
188. */
189. @Test
190. public void checkTest10(){
191.     psw = "ingsw";
192.     expResult = false;
193.     result = instance.check(user, psw);
```

```
194.     assertEquals(result, expResult);
195. }
196.
197. /*
198. * EQUIVALENCE CLASSES TESTED:
199. * User: F4
200. * Psw: F1
201. */
202. @Test
203. public void checkTest11(){
204.     psw = "ingw";
205.     expResult = false;
206.     result = instance.check(user, psw);
207.     assertEquals(result, expResult);
208. }
209.
210. /*
211. * EQUIVALENCE CLASSES TESTED:
212. * User: F4
213. * Psw: F2
214. */
215. @Test
216. public void checkTest12(){
217.     expResult = false;
218.     result = instance.check(user, psw);
219.     assertEquals(result, expResult);
220. }
221.
222. /*
223. * EQUIVALENCE CLASSES TESTED:
224. * User: F1
225. * Psw: T
226. */
227. @Test
228. public void checkTest13(){
229.     user = "-1";
230.     psw = "ingsw";
231.     expResult = false;
232.     result = instance.check(user, psw);
233.     assertEquals(result, expResult);
234. }
235.
236. /*
237. * EQUIVALENCE CLASSES TESTED:
238. * User: F1
239. * Psw: F1
240. */
241. @Test
242. public void checkTest14(){
243.     user = "-1";
244.     psw = "ingw";
245.     expResult = false;
246.     result = instance.check(user, psw);
247.     assertEquals(result, expResult);
248. }
249.
250. /*
251. * EQUIVALENCE CLASSES TESTED:
252. * User: F1
253. * Psw: F2
254. */
255. @Test
256. public void checkTest15(){
257.     user = "-1";
258.     expResult = false;
259.     result = instance.check(user, psw);
```

```
260.     assertEquals(result, expResult);
261. }
262. }
263.
```

Results

Tests passed: 100,00 %

All 15 tests passed. (2,546 s)

- └ Controller.Login_ControllerTest passed
 - checkTest1 passed (0,047 s)
 - checkTest2 passed (0,031 s)
 - checkTest3 passed (0,047 s)
 - checkTest4 passed (0,031 s)
 - checkTest5 passed (0,047 s)
 - checkTest6 passed (0,032 s)
 - checkTest7 passed (0,0 s)
 - checkTest8 passed (0,0 s)
 - checkTest9 passed (0,0 s)
 - checkTest10 passed (0,0 s)
 - checkTest11 passed (0,0 s)
 - checkTest12 passed (0,0 s)
 - checkTest13 passed (0,0 s)
 - checkTest14 passed (0,0 s)
 - checkTest15 passed (0,0 s)

Testsuite: Controller.Login_ControllerTest

Tests run: 15, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2,546 sec