

FactDev

0.1

Generated by Doxygen 1.8.8

Wed Feb 4 2015 00:17:36



# Contents

<b>1</b>	<b>FactDev documentation</b>	<b>1</b>
<b>2</b>	<b>QTestRunner</b>	<b>3</b>
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Class Index</b>	<b>7</b>
4.1	Class List . . . . .	7
<b>5</b>	<b>Class Documentation</b>	<b>11</b>
5.1	AddContributoryDialog Class Reference . . . . .	11
5.1.1	Detailed Description . . . . .	11
5.1.2	Constructor & Destructor Documentation . . . . .	11
5.1.2.1	AddContributoryDialog . . . . .	11
5.2	AddProjectDialog Class Reference . . . . .	12
5.2.1	Detailed Description . . . . .	12
5.2.2	Constructor & Destructor Documentation . . . . .	13
5.2.2.1	AddProjectDialog . . . . .	13
5.2.2.2	AddProjectDialog . . . . .	14
5.3	AddQuoteDialog Class Reference . . . . .	14
5.3.1	Detailed Description . . . . .	14
5.3.2	Constructor & Destructor Documentation . . . . .	15
5.3.2.1	AddQuoteDialog . . . . .	15
5.4	Billing Class Reference . . . . .	15
5.4.1	Detailed Description . . . . .	16
5.4.2	Constructor & Destructor Documentation . . . . .	16
5.4.2.1	Billing . . . . .	16
5.4.3	Member Function Documentation . . . . .	16
5.4.3.1	addContributory . . . . .	16
5.4.3.2	getContributories . . . . .	17
5.4.3.3	getDate . . . . .	17
5.4.3.4	getDescription . . . . .	17

5.4.3.5	<a href="#">getNumber</a>	17
5.4.3.6	<a href="#">getTitle</a>	17
5.4.3.7	<a href="#">hydrat</a>	17
5.4.3.8	<a href="#">isBilling</a>	18
5.4.3.9	<a href="#">operator"!="</a>	18
5.4.3.10	<a href="#">operator=="</a>	18
5.4.3.11	<a href="#">setDate</a>	18
5.4.3.12	<a href="#">setDescription</a>	18
5.4.3.13	<a href="#">setIsBilling</a>	19
5.4.3.14	<a href="#">setNumber</a>	19
5.4.3.15	<a href="#">setTitle</a>	19
5.5	<a href="#">BillingDatabase Class Reference</a>	19
5.5.1	<a href="#">Detailed Description</a>	20
5.5.2	<a href="#">Member Function Documentation</a>	20
5.5.2.1	<a href="#">addBilling</a>	20
5.5.2.2	<a href="#">addBillingProject</a>	20
5.5.2.3	<a href="#">getBilling</a>	20
5.5.2.4	<a href="#">getBillingsTable</a>	21
5.5.2.5	<a href="#">instance</a>	21
5.5.2.6	<a href="#">removeBilling</a>	21
5.6	<a href="#">BillingDatabaseTest Class Reference</a>	21
5.7	<a href="#">BillingModelTest Class Reference</a>	22
5.8	<a href="#">CheckCity Class Reference</a>	22
5.8.1	<a href="#">Detailed Description</a>	23
5.8.2	<a href="#">Constructor &amp; Destructor Documentation</a>	23
5.8.2.1	<a href="#">CheckCity</a>	23
5.9	<a href="#">CheckCountry Class Reference</a>	23
5.9.1	<a href="#">Detailed Description</a>	23
5.9.2	<a href="#">Constructor &amp; Destructor Documentation</a>	24
5.9.2.1	<a href="#">CheckCountry</a>	24
5.10	<a href="#">CheckEmail Class Reference</a>	25
5.10.1	<a href="#">Detailed Description</a>	25
5.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	25
5.10.2.1	<a href="#">CheckEmail</a>	25
5.10.3	<a href="#">Member Function Documentation</a>	25
5.10.3.1	<a href="#">check</a>	26
5.11	<a href="#">CheckFieldsLetters Class Reference</a>	26
5.11.1	<a href="#">Detailed Description</a>	27
5.11.2	<a href="#">Constructor &amp; Destructor Documentation</a>	27
5.11.2.1	<a href="#">CheckFieldsLetters</a>	27

5.11.3	Member Function Documentation	27
5.11.3.1	check	27
5.12	CheckName Class Reference	27
5.12.1	Detailed Description	28
5.12.2	Constructor & Destructor Documentation	28
5.12.2.1	CheckName	28
5.13	CheckPhone Class Reference	28
5.13.1	Detailed Description	29
5.13.2	Constructor & Destructor Documentation	29
5.13.2.1	CheckPhone	29
5.13.3	Member Function Documentation	29
5.13.3.1	check	29
5.13.3.2	getCountry	29
5.13.3.3	setCountry	29
5.14	CheckPostalCode Class Reference	30
5.14.1	Detailed Description	30
5.14.2	Constructor & Destructor Documentation	30
5.14.2.1	CheckPostalCode	30
5.14.3	Member Function Documentation	30
5.14.3.1	check	30
5.14.3.2	getCountry	31
5.14.3.3	setCountry	31
5.15	CheckQLineEdit Class Reference	31
5.15.1	Detailed Description	32
5.15.2	Constructor & Destructor Documentation	32
5.15.2.1	CheckQLineEdit	32
5.15.3	Member Function Documentation	32
5.15.3.1	getBtnValid	32
5.15.3.2	isValid	32
5.15.3.3	setBtnValid	32
5.16	CheckSiretNumber Class Reference	33
5.16.1	Detailed Description	33
5.16.2	Constructor & Destructor Documentation	33
5.16.2.1	CheckSiretNumber	33
5.16.3	Member Function Documentation	33
5.16.3.1	check	34
5.17	CheckUntilField Class Reference	35
5.17.1	Detailed Description	35
5.17.2	Constructor & Destructor Documentation	35
5.17.2.1	CheckUntilField	35

5.17.3 Member Function Documentation . . . . .	36
5.17.3.1 check . . . . .	36
5.18 CheckValidField Class Reference . . . . .	36
5.18.1 Detailed Description . . . . .	36
5.18.2 Constructor & Destructor Documentation . . . . .	37
5.18.2.1 CheckValidField . . . . .	37
5.18.3 Member Function Documentation . . . . .	37
5.18.3.1 check . . . . .	37
5.19 ComboBoxModelWidget Class Reference . . . . .	37
5.19.1 Detailed Description . . . . .	37
5.20 ContributoriesDatabaseTest Class Reference . . . . .	38
5.21 ContributoriesTableModel Class Reference . . . . .	38
5.21.1 Detailed Description . . . . .	39
5.21.2 Constructor & Destructor Documentation . . . . .	39
5.21.2.1 ContributoriesTableModel . . . . .	39
5.21.3 Member Function Documentation . . . . .	39
5.21.3.1 append . . . . .	39
5.21.3.2 columnCount . . . . .	39
5.21.3.3 count . . . . .	39
5.21.3.4 data . . . . .	39
5.21.3.5 flags . . . . .	40
5.21.3.6 getContributories . . . . .	40
5.21.3.7 headerData . . . . .	40
5.21.3.8 remove . . . . .	40
5.21.3.9 rowCount . . . . .	40
5.21.3.10 setData . . . . .	41
5.22 ContributoriesWidget Class Reference . . . . .	41
5.22.1 Member Function Documentation . . . . .	42
5.22.1.1 getContributories . . . . .	42
5.23 Contributory Class Reference . . . . .	42
5.23.1 Detailed Description . . . . .	43
5.23.2 Constructor & Destructor Documentation . . . . .	43
5.23.2.1 Contributory . . . . .	43
5.23.3 Member Function Documentation . . . . .	43
5.23.3.1 getDescription . . . . .	43
5.23.3.2 getNbHours . . . . .	43
5.23.3.3 getProject . . . . .	43
5.23.3.4 hydrat . . . . .	43
5.23.3.5 setDescription . . . . .	44
5.23.3.6 setNbHours . . . . .	44

5.23.3.7	setProject	44
5.24	ContributoryDatabase Class Reference	44
5.24.1	Detailed Description	45
5.24.2	Member Function Documentation	45
5.24.2.1	addContributory	45
5.24.2.2	getContributory	45
5.24.2.3	getContributory	45
5.24.2.4	instance	46
5.24.2.5	removeContributory	46
5.25	Customer Class Reference	46
5.25.1	Detailed Description	48
5.25.2	Constructor & Destructor Documentation	48
5.25.2.1	Customer	48
5.25.3	Member Function Documentation	48
5.25.3.1	getAddress	48
5.25.3.2	getCity	48
5.25.3.3	getCompany	48
5.25.3.4	getCountry	49
5.25.3.5	getEmail	49
5.25.3.6	getFax	49
5.25.3.7	getFirstnameReferent	49
5.25.3.8	getLastnameReferent	49
5.25.3.9	getMobilePhone	49
5.25.3.10	getPhone	49
5.25.3.11	getPostalCode	50
5.25.3.12	hydrat	50
5.25.3.13	operator"!="	50
5.25.3.14	operator=="	50
5.25.3.15	setAddress	50
5.25.3.16	setCity	50
5.25.3.17	setCompany	51
5.25.3.18	setCountry	51
5.25.3.19	setEmail	51
5.25.3.20	setFax	51
5.25.3.21	setFirstnameReferent	51
5.25.3.22	setLastnameReferent	51
5.25.3.23	setMobilePhone	52
5.25.3.24	setPhone	52
5.25.3.25	setPostalCode	52
5.26	CustomerContextualMenu Class Reference	52

5.26.1 Detailed Description . . . . .	53
5.26.2 Constructor & Destructor Documentation . . . . .	53
5.26.2.1 CustomerContextualMenu . . . . .	53
5.27 CustomerDatabase Class Reference . . . . .	53
5.27.1 Detailed Description . . . . .	54
5.27.2 Member Function Documentation . . . . .	54
5.27.2.1 addCustomer . . . . .	54
5.27.2.2 getCustomer . . . . .	54
5.27.2.3 getCustomersTable . . . . .	54
5.27.2.4 getCustomersTree . . . . .	55
5.27.2.5 getNbCustomers . . . . .	55
5.27.2.6 instance . . . . .	55
5.27.2.7 removeCustomer . . . . .	55
5.28 CustomerDatabaseTest Class Reference . . . . .	56
5.29 CustomerDataWidget Class Reference . . . . .	56
5.29.1 Detailed Description . . . . .	56
5.29.2 Member Function Documentation . . . . .	57
5.29.2.1 printInformations . . . . .	57
5.30 CustomerModelTest Class Reference . . . . .	57
5.31 Database Class Reference . . . . .	57
5.31.1 Detailed Description . . . . .	59
5.31.2 Member Function Documentation . . . . .	59
5.31.2.1 executeFile . . . . .	59
5.31.2.2 instance . . . . .	59
5.31.2.3 lastError . . . . .	59
5.31.2.4 setDatabase . . . . .	59
5.31.2.5 value . . . . .	59
5.32 DbException Class Reference . . . . .	60
5.32.1 Detailed Description . . . . .	60
5.32.2 Constructor & Destructor Documentation . . . . .	60
5.32.2.1 DbException . . . . .	60
5.32.3 Member Function Documentation . . . . .	61
5.32.3.1 popupMessage . . . . .	61
5.33 DialogAddCustomer Class Reference . . . . .	61
5.33.1 Detailed Description . . . . .	62
5.33.2 Constructor & Destructor Documentation . . . . .	62
5.33.2.1 DialogAddCustomer . . . . .	62
5.34 ICheckField Class Reference . . . . .	62
5.34.1 Detailed Description . . . . .	62
5.34.2 Member Function Documentation . . . . .	63



5.34.2.1	check	63
5.35	IDatabaseModel Class Reference	63
5.35.1	Detailed Description	64
5.35.2	Member Function Documentation	64
5.35.2.1	getId	64
5.35.2.2	hydrat	64
5.35.2.3	isToRemoved	64
5.35.2.4	setId	64
5.35.2.5	setToRemoved	64
5.36	ItemType Class Reference	65
5.36.1	Detailed Description	65
5.36.2	Constructor & Destructor Documentation	65
5.36.2.1	ItemType	65
5.36.3	Member Function Documentation	66
5.36.3.1	getModel	66
5.36.3.2	getName	66
5.36.3.3	getType	66
5.36.3.4	setName	66
5.36.3.5	setType	66
5.37	Log Class Reference	67
5.37.1	Detailed Description	67
5.37.2	Member Function Documentation	67
5.37.2.1	instance	67
5.37.2.2	write	67
5.37.3	Friends And Related Function Documentation	68
5.37.3.1	operator<<	68
5.38	MainWindow Class Reference	68
5.38.1	Detailed Description	69
5.38.2	Constructor & Destructor Documentation	69
5.38.2.1	MainWindow	69
5.38.3	Member Function Documentation	70
5.38.3.1	addQuote	70
5.38.3.2	editUser	70
5.38.3.3	getCurrentCustomerId	70
5.38.3.4	getCurrentCustomerName	70
5.38.3.5	getCurrentProjectId	70
5.38.3.6	getCurrentProjectName	70
5.38.3.7	isCustomerItemTree	71
5.38.3.8	isProjectItemTree	71
5.38.3.9	isTreeRoot	71

5.38.3.10 newProject . . . . .	71
5.38.3.11 search . . . . .	71
5.39 MessageBox Class Reference . . . . .	71
5.39.1 Detailed Description . . . . .	72
5.39.2 Constructor & Destructor Documentation . . . . .	72
5.39.2.1 MessageBox . . . . .	72
5.39.3 Member Function Documentation . . . . .	72
5.39.3.1 setImage . . . . .	72
5.39.3.2 setText . . . . .	73
5.40 Parameters Class Reference . . . . .	73
5.40.1 Member Data Documentation . . . . .	73
5.40.1.1 DB_FILENAME . . . . .	73
5.40.1.2 VERSION . . . . .	73
5.41 Popup Class Reference . . . . .	74
5.41.1 Detailed Description . . . . .	74
5.42 Project Class Reference . . . . .	74
5.42.1 Detailed Description . . . . .	75
5.42.2 Constructor & Destructor Documentation . . . . .	75
5.42.2.1 Project . . . . .	75
5.42.3 Member Function Documentation . . . . .	76
5.42.3.1 getBeginDate . . . . .	76
5.42.3.2 getCustomer . . . . .	76
5.42.3.3 getDailyRate . . . . .	76
5.42.3.4 getDescription . . . . .	76
5.42.3.5 getEndDate . . . . .	76
5.42.3.6 getName . . . . .	76
5.42.3.7 hydrat . . . . .	77
5.42.3.8 setBeginDate . . . . .	78
5.42.3.9 setCustomer . . . . .	78
5.42.3.10 setDailyRate . . . . .	78
5.42.3.11 setDescription . . . . .	78
5.42.3.12 setEndDate . . . . .	78
5.42.3.13 setName . . . . .	78
5.43 ProjectComboDelegate Class Reference . . . . .	79
5.44 ProjectDatabase Class Reference . . . . .	79
5.44.1 Detailed Description . . . . .	80
5.44.2 Member Function Documentation . . . . .	80
5.44.2.1 addProject . . . . .	80
5.44.2.2 getNbProjects . . . . .	80
5.44.2.3 getNbProjectsForACustomer . . . . .	81

5.44.2.4	<a href="#">getProject</a>	82
5.44.2.5	<a href="#">getProject</a>	82
5.44.2.6	<a href="#">getProjectsOfCustomer</a>	82
5.44.2.7	<a href="#">getProjectsTable</a>	82
5.44.2.8	<a href="#">instance</a>	83
5.44.2.9	<a href="#">removeProject</a>	83
5.45	<a href="#">ProjectsWidget Class Reference</a>	83
5.45.1	<a href="#">Detailed Description</a>	84
5.45.2	<a href="#">Constructor &amp; Destructor Documentation</a>	84
5.45.2.1	<a href="#">ProjectsWidget</a>	84
5.46	<a href="#">RateWidget Class Reference</a>	84
5.46.1	<a href="#">Detailed Description</a>	85
5.46.2	<a href="#">Constructor &amp; Destructor Documentation</a>	85
5.46.2.1	<a href="#">RateWidget</a>	85
5.46.3	<a href="#">Member Function Documentation</a>	85
5.46.3.1	<a href="#">getDailyRate</a>	85
5.46.3.2	<a href="#">getHourlyRate</a>	85
5.46.3.3	<a href="#">setDailyRate</a>	85
5.46.3.4	<a href="#">setHourlyRate</a>	86
5.46.3.5	<a href="#">setWidgetDailyRateValue</a>	86
5.46.3.6	<a href="#">setWidgetHourlyRateValue</a>	86
5.47	<a href="#">Search Class Reference</a>	86
5.47.1	<a href="#">Detailed Description</a>	87
5.47.2	<a href="#">Member Function Documentation</a>	87
5.47.2.1	<a href="#">getFilter</a>	87
5.47.2.2	<a href="#">getGroupFilter</a>	87
5.47.2.3	<a href="#">getSearchInCompanies</a>	87
5.47.2.4	<a href="#">getSearchInReferentLastname</a>	87
5.47.2.5	<a href="#">getText</a>	88
5.47.2.6	<a href="#">setGroupFilter</a>	88
5.47.2.7	<a href="#">setSearchInCompanies</a>	88
5.47.2.8	<a href="#">setSearchInReferentLastname</a>	88
5.47.2.9	<a href="#">setText</a>	88
5.48	<a href="#">searchTest Class Reference</a>	88
5.49	<a href="#">searchWidget Class Reference</a>	89
5.49.1	<a href="#">Detailed Description</a>	89
5.49.2	<a href="#">Constructor &amp; Destructor Documentation</a>	90
5.49.2.1	<a href="#">searchWidget</a>	90
5.49.3	<a href="#">Member Function Documentation</a>	90
5.49.3.1	<a href="#">getCurrentCustomerId</a>	90

5.49.3.2	isCustomerSelected	90
5.49.3.3	search	90
5.50	TestAdder< T > Class Template Reference	90
5.51	testadder Class Reference	91
5.52	TestRunner Class Reference	91
5.53	User Class Reference	91
5.53.1	Detailed Description	92
5.53.2	Constructor & Destructor Documentation	93
5.53.2.1	User	93
5.53.3	Member Function Documentation	93
5.53.3.1	getAddress	93
5.53.3.2	getCity	93
5.53.3.3	getCompany	93
5.53.3.4	getEmail	93
5.53.3.5	getFirstname	93
5.53.3.6	getLastname	94
5.53.3.7	getMobilePhone	94
5.53.3.8	getNoSiret	94
5.53.3.9	getPhone	94
5.53.3.10	getPostalCode	94
5.53.3.11	getTitle	94
5.53.3.12	hydrat	94
5.53.3.13	setAddress	95
5.53.3.14	setCity	95
5.53.3.15	setCompany	95
5.53.3.16	setEmail	95
5.53.3.17	setFirstname	95
5.53.3.18	setLastname	95
5.53.3.19	setMobilePhone	96
5.53.3.20	setNoSiret	96
5.53.3.21	setPhone	96
5.53.3.22	setPostalCode	96
5.53.3.23	setTitle	96
5.54	UserDataBase Class Reference	96
5.54.1	Detailed Description	97
5.54.2	Member Function Documentation	97
5.54.2.1	getUser	97
5.54.2.2	getUserTable	98
5.54.2.3	instance	98
5.55	UserDataDialog Class Reference	98

---

5.55.1 Detailed Description . . . . .	99
5.55.2 Constructor & Destructor Documentation . . . . .	99
5.55.2.1 UserDataDialog . . . . .	99
5.56 Utils Class Reference . . . . .	99
5.56.1 Detailed Description . . . . .	99
5.56.2 Member Function Documentation . . . . .	99
5.56.2.1 firstLetterToUpper . . . . .	99
<b>Index</b>	<b>101</b>



# Chapter 1

## FactDev documentation

This website contains the documentation of FactDev, a billings and quotes software.

### Team

FACT team is a team for University [Project](#). Members of this team are :

- Florent Berbie ([KraTuX31](#))
- Antoine de Roquemaurel ([aroquemaurel](#))
- Cédric Rohaut ([Oxynos](#))
- Manantsoa Andriamihary Razanajatovo ([manantsoa](#))

For more informations, you can go to <http://fact-team.github.io>.

### What is FactDev ?

FactDev is a software for quotes and billings developped by FACT team for university project in Toulouse II↔ I University – Paul Sabatier.

This software is developed with C++ and Qt framework, and there is differents features :

- Customers database
- Differents projects for customers
- Quotes
- Billings

### Documentation

You can access to the documentation generated by Doxygen here :

- [HTML Documentation](#)
- [PDF Documentation](#)

## Installation and using

- [User Manual](#)



## Chapter 2

# QTestRunner

A repo for Increase QTest productivity

Thanks to <https://marcoarena.wordpress.com/2012/06/23/increase-your-qtest-productivity/>!

The QTestLib framework is a tool for unit testing Qt based applications and libraries. I find it precious and simple to use, though it lacks some important features, supported, for example, by GTest. I'm not talking about mocking – for this you generally need an out-and-out framework (like GMock) – instead, I'm referring to simple things like fast deploying. Suppose you have written a test class like this:

This repo contains only the code in this article.



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Database . . . . .	??
BillingDatabase . . . . .	??
ContributoryDatabase . . . . .	??
CustomerDatabase . . . . .	??
ProjectDatabase . . . . .	??
UserDatabase . . . . .	??
exception	
DbException . . . . .	??
ICheckField . . . . .	??
CheckQLEdit . . . . .	??
CheckEmail . . . . .	??
CheckSiretNumber . . . . .	??
CheckUntilField . . . . .	??
CheckFieldsLetters . . . . .	??
CheckCity . . . . .	??
CheckCountry . . . . .	??
CheckName . . . . .	??
CheckPhone . . . . .	??
CheckPostalCode . . . . .	??
CheckValidField . . . . .	??
IDatabaseModel . . . . .	??
Billing . . . . .	??
Contributory . . . . .	??
Customer . . . . .	??
Project . . . . .	??
User . . . . .	??
ItemType . . . . .	??
Log . . . . .	??
Parameters . . . . .	??
Popup . . . . .	??
QAbstractTableModel	
ContributoriesTableModel . . . . .	??
QDialog	
AddContributoryDialog . . . . .	??
AddProjectDialog . . . . .	??
AddQuoteDialog . . . . .	??
DialogAddCustomer . . . . .	??

MessageBox . . . . .	??
UserDataDialog . . . . .	??
QItemDelegate	
ProjectComboDelegate . . . . .	??
QLineEdit	
CheckQLineEdit . . . . .	??
QMainWindow	
MainWindow . . . . .	??
QMenu	
CustomerContextualMenu . . . . .	??
QObject	
BillingDatabaseTest . . . . .	??
BillingModelTest . . . . .	??
ContributoriesDatabaseTest . . . . .	??
CustomerDatabaseTest . . . . .	??
CustomerModelTest . . . . .	??
searchTest . . . . .	??
QWidget	
ComboBoxModelWidget . . . . .	??
ContributoriesWidget . . . . .	??
CustomerDataWidget . . . . .	??
ProjectsWidget . . . . .	??
RateWidget . . . . .	??
searchWidget . . . . .	??
Search . . . . .	??
TestAdder< T > . . . . .	??
testadder . . . . .	??
TestRunner . . . . .	??
Utils . . . . .	??

## Chapter 4

# Class Index

### 4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">AddContributoryDialog</a>	Windows to add a new <a href="#">Contributory</a>	??
<a href="#">AddProjectDialog</a>	Windows to add a new <a href="#">Project</a>	??
<a href="#">AddQuoteDialog</a>	Window to add or modify a Quote	??
<a href="#">Billing</a>	: <a href="#">Billing</a> or Quote of a <a href="#">Customer</a>	??
<a href="#">BillingDatabase</a>	The <a href="#">BillingDatabase</a> class <a href="#">Billing</a> (or Quote) table database	??
<a href="#">BillingDatabaseTest</a>		??
<a href="#">BillingModelTest</a>		??
<a href="#">CheckCity</a>	Line Edit of City with a check icon	??
<a href="#">CheckCountry</a>	<a href="#">CheckCountry::CheckCountry</a> Line Edit of country with a check icon	??
<a href="#">CheckEmail</a>	Line Edit of email with a check icon	??
<a href="#">CheckFieldsLetters</a>	Field with only letters (no numbers)	??
<a href="#">CheckName</a>	Line edit of name with a check icon	??
<a href="#">CheckPhone</a>	Line Edit of Phone number with a check icon	??
<a href="#">CheckPostalCode</a>	Line Edit of postal code with a check icon	??
<a href="#">CheckQLineEdit</a>	LineEdit custom with a check of text inputed	??
<a href="#">CheckSiretNumber</a>	Line Edit with a check icon	??
<a href="#">CheckUntilField</a>	The <a href="#">CheckUntilField</a> class	??
<a href="#">CheckValidField</a>	Check field not required	??
<a href="#">ComboBoxModelWidget</a>	The <a href="#">ComboBoxModelWidget</a> class TODO	??
<a href="#">ContributoriesDatabaseTest</a>		??

<a href="#">ContributoriesTableModel</a>	
For a custom table for contributories widget	??
<a href="#">ContributoriesWidget</a>	??
<a href="#">Contributory</a>	
The <a href="#">Contributory</a> class	??
<a href="#">ContributoryDatabase</a>	
The <a href="#">ContributoryDatabase</a> class <a href="#">Contributory</a> (or Quote) table database	??
<a href="#">Customer</a>	
<a href="#">Customer</a>	??
<a href="#">CustomerContextualMenu</a>	
Display contextual menu on a customer	??
<a href="#">CustomerDatabase</a>	
The <a href="#">CustomerDatabase</a> class <a href="#">Customer</a> table database	??
<a href="#">CustomerDatabaseTest</a>	??
<a href="#">CustomerDataWidget</a>	
Class for display info of a customer	??
<a href="#">CustomerModelTest</a>	??
<a href="#">Database</a>	
The <a href="#">Database</a> class Master class for all database access	??
<a href="#">DbException</a>	
For database exception : queries, db file, ...	??
<a href="#">DialogAddCustomer</a>	
Window to add or modify a <a href="#">Customer</a>	??
<a href="#">ICheckField</a>	
Interface to check fields validity	??
<a href="#">IDatabaseModel</a>	
The <a href="#">IDatabaseModel</a> class	??
<a href="#">ItemType</a>	
Item type model	??
<a href="#">Log</a>	
For Simple management of log	??
<a href="#">MainWindow</a>	
Main Window of the software	??
<a href="#">MessageBox</a>	
Information window with message	??
<a href="#">Parameters</a>	??
<a href="#">Popup</a>	
Class for display popup quickly	??
<a href="#">Project</a>	
: <a href="#">Project</a> linked to a <a href="#">Customer</a>	??
<a href="#">ProjectComboDelegate</a>	??
<a href="#">ProjectDatabase</a>	
<a href="#">Project</a> table database	??
<a href="#">ProjectsWidget</a>	
Actions on <a href="#">Project</a>	??
<a href="#">RateWidget</a>	
Class for display Rate	??
<a href="#">Search</a>	
The <a href="#">Search</a> class	??
<a href="#">searchTest</a>	??
<a href="#">searchWidget</a>	
Class for search in database	??
<a href="#">TestAdder&lt; T &gt;</a>	??
<a href="#">testadder</a>	??
<a href="#">TestRunner</a>	??
<a href="#">User</a>	
<a href="#">User</a> of it application	??

<a href="#">UserDatabase</a>	Access to <a href="#">User</a> data in the the table <a href="#">User</a> of the <a href="#">Database</a> . . . . .	??
<a href="#">UserDataDialog</a>	Window to fill user data . . . . .	??
<a href="#">Utils</a>	The <a href="#">Utils</a> class . . . . .	??





## Chapter 5

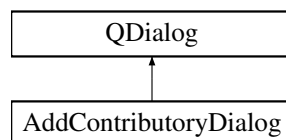
# Class Documentation

### 5.1 AddContributoryDialog Class Reference

The [AddContributoryDialog](#) class Windows to add a new [Contributory](#).

```
#include <addcontributorydialog.h>
```

Inheritance diagram for AddContributoryDialog:



#### Public Member Functions

- [AddContributoryDialog](#) (QWidget \*parent=0)  
*[AddContributoryDialog::AddContributoryDialog](#) Construct a new windows AddContributory.*

#### 5.1.1 Detailed Description

The [AddContributoryDialog](#) class Windows to add a new [Contributory](#).

Author

Florent Berbie

See also

[Contributory](#)

#### 5.1.2 Constructor & Destructor Documentation

5.1.2.1 [AddContributoryDialog::AddContributoryDialog](#) ( [QWidget](#) \* *parent* = 0 ) [explicit]

[AddContributoryDialog::AddContributoryDialog](#) Construct a new windows AddContributory.

## Parameters

<i>parent</i>	QWidget
---------------	---------

The documentation for this class was generated from the following files:

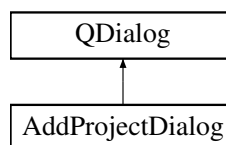
- `src/dialogs/addcontributorydialog.h`
- `src/dialogs/addcontributorydialog.cpp`

## 5.2 AddProjectDialog Class Reference

The [AddProjectDialog](#) class Windows to add a new [Project](#).

```
#include <addprojectdialog.h>
```

Inheritance diagram for AddProjectDialog:



### Public Slots

- void [checkFields](#) ()  
[AddProjectDialog::checkFields](#) Check if fields are valid.

### Public Member Functions

- [AddProjectDialog](#) (int id=0, QWidget \*parent=0)  
[AddProjectDialog::AddProjectDialog](#) Construct a windows [AddProjectDialog](#).
- [AddProjectDialog](#) (int noRowCustomer, int idProject=0, QWidget \*parent=0)  
[AddProjectDialogAddProjectDialog](#) Construct a windows according an idCustomer and, optionnaly, an idProject
- void [accept](#) ()  
[AddProjectDialog::accept](#) Valid data inputed by user and add these data in [Database](#).
- void [reject](#) ()  
[AddProjectDialog::reject](#) Cancel the operation and close the windows.
- void [fillFields](#) ()  
[AddProjectDialog::fillFields](#) Fill the differents fields of the current windows according the [Project](#) data existing As a project requires to be linked to a [Customer](#), the [Customer](#) selection part may be disable.

### 5.2.1 Detailed Description

The [AddProjectDialog](#) class Windows to add a new [Project](#).

#### Author

Florent Berbie

#### See also

[Project](#)

## 5.2.2 Constructor & Destructor Documentation

5.2.2.1 AddProjectDialog::AddProjectDialog ( int *id* = 0, QWidget \* *parent* = 0 ) [explicit]

[AddProjectDialog::AddProjectDialog](#) Construct a windows [AddProjectDialog](#).

## Parameters

<i>id</i>	<a href="#">Project</a> identity
<i>parent</i>	QWidget of the current windows

### 5.2.2.2 AddProjectDialog::AddProjectDialog ( int *noRowCustomer*, int *idProject* = 0, QWidget \* *parent* = 0 ) [explicit]

AddProjectDialogAddProjectDialog Construct a windows according an *idCustomer* and, optionnaly, an *idProject*

## Parameters

<i>noRowCustomer</i>	Row number of the <a href="#">Customer</a>
<i>idProject</i>	<a href="#">Project</a> identify
<i>parent</i>	QWidget of the current windows

The documentation for this class was generated from the following files:

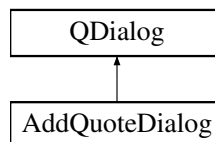
- src/dialogs/addprojectdialog.h
- src/dialogs/addprojectdialog.cpp

## 5.3 AddQuoteDialog Class Reference

The [AddQuoteDialog](#) class Window to add or modify a Quote.

```
#include <addquotedialog.h>
```

Inheritance diagram for AddQuoteDialog:



## Public Slots

- void **updateBtn** (void)

## Public Member Functions

- [AddQuoteDialog](#) (int *idCustomer*=0, int *id*=0, QWidget \**parent*=0)  
*AddQuoteDialog::AddQuoteDialog* Construct a windows [AddQuoteDialog](#).
- void [fillFields](#) ()  
*AddQuoteDialog::fillFields* Fill line edits with the data of the quote.
- void [accept](#) ()  
*AddQuoteDialog::accept* Valid data inputed by user and add these data in [Database](#).
- void [reject](#) ()  
*AddQuoteDialog::reject* Cancel the operation and close the windows.

### 5.3.1 Detailed Description

The [AddQuoteDialog](#) class Window to add or modify a Quote.

Author

Cédric Rohaut

### 5.3.2 Constructor & Destructor Documentation

5.3.2.1 `AddQuoteDialog::AddQuoteDialog ( int idCustomer = 0, int id = 0, QWidget * parent = 0 )` `[explicit]`

`AddQuoteDialog::AddQuoteDialog` Construct a windows `AddQuoteDialog`.

Parameters

<i>parent</i>	QWidget of the current windows
---------------	--------------------------------

The documentation for this class was generated from the following files:

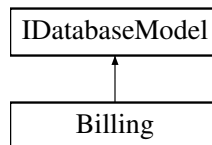
- `src/dialogs/addquotedialog.h`
- `src/dialogs/addquotedialog.cpp`

## 5.4 Billing Class Reference

The `Billing` class : `Billing` or Quote of a `Customer`.

```
#include <billing.h>
```

Inheritance diagram for `Billing`:



### Public Member Functions

- `Billing ()`  
*Billing::Billing.* Construct a `Billing`.
- `Billing (int id)`  
*Billing::Billing.* Construct a `Billing` or quote.
- `~Billing ()`  
*destruct a billing object*
- `void commit ()`  
*Billing::commit.* Insert a modification in `Billing` table on the database.
- `void hydrat (int getId)`  
*Billing::hydrat.* Update of the `Billing` which is specified by `getId`
- `void remove ()`  
*Billing::remove.* Remove a `Billing`.
- `QMap< Project *, QList< Contributory > > getContributories () const`  
*Billing::getContributories.* Return a map of `Contributory` for each `Project` of the `Billing`
- `void setContributories (QMap< Project *, QList< Contributory > > contributories)`
- `void addContributory (Contributory &c)`  
*addContributories* Add a new contributory for project *p*
- `QString getTitle () const`

- *Billing::getTitle.* return title of **Billing**
- void **setTitle** (const QString &**getTitle**)
  - Billing::setTitle.* Modify the title of **Billing**
- QString **getDescription** () const
  - Billing::getDescription.* return description of **Billing**
- void **setDescription** (const QString &**getDescription**)
  - Billing::setDescription.* Modify the description of **Billing**
- int **getNumber** () const
  - Billing::getNumber.* Return number of the **Billing**.
- void **setNumber** (int **getNumber**)
  - Billing::setNumber.* Modify *\_number* of **Billing**.
- bool **isBilling** () const
  - Billing::isBilling.* Return if it's a billing or a quote.
- void **setIsBilling** (bool **isBilling**)
  - Billing::setIsBilling.* Modify *isBilling* of **Billing**.
- QDate **getDate** () const
  - Billing::getDate.* return date of the **Billing**
- void **setDate** (const QDate &**getDate**)
  - Billing::setDate.* Modify date of the **Billing**
- bool **operator==** (const **Billing** &b)
  - Billing::operator ==* define the operator "==" to compare two billings and to see if they are the same.
- bool **operator!=** (const **Billing** &b)
  - Billing::operator !=* defines the operator "!=" to compare two **Billing** and to see if they are different.

## Additional Inherited Members

### 5.4.1 Detailed Description

The **Billing** class : **Billing** or Quote of a **Customer**.

Author

Cédric Rohaut for the quote part

### 5.4.2 Constructor & Destructor Documentation

#### 5.4.2.1 **Billing::Billing** ( int *id* )

**Billing::Billing.** Construct a **Billing** or quote.

Parameters

<i>int</i>	<i>id</i>
------------	-----------

### 5.4.3 Member Function Documentation

#### 5.4.3.1 void **Billing::addContributory** ( **Contributory** & c )

**addContributories** Add a new contributory for project p

## Parameters

<i>p</i>	The <a href="#">Project</a> who contain <a href="#">Contributory</a>
<i>c</i>	The new <a href="#">Contributory</a>

5.4.3.2 QMap< [Project](#) \*, QList< [Contributory](#) > > Billing::getContributories ( ) const

[Billing::getContributories](#). Return a map of [Contributory](#) for each [Project](#) of the [Billing](#)

## Returns

QMap<[Project](#), QList<[Contributory](#)>>

5.4.3.3 QDate Billing::getDate ( ) const

[Billing::getDate](#). return date of the [Billing](#)

## Returns

date of [Billing](#)

5.4.3.4 QString Billing::getDescription ( ) const

[Billing::getDescription](#). return description of [Billing](#)

## Returns

description of [Billing](#)

5.4.3.5 int Billing::getNumber ( ) const

[Billing::getNumber](#). Return number of the [Billing](#).

## Returns

\_number of [Billing](#)

5.4.3.6 QString Billing::getTitle ( ) const

[Billing::getTitle](#). return title of [Billing](#)

## Returns

title of [Billing](#)

5.4.3.7 void Billing::hydrat ( int *getId* ) [virtual]

[Billing::hydrat](#). Update of the [Billing](#) which is specified by *getId*

## Parameters

<i>getId</i>	
--------------	--

Implements [IDatabaseModel](#).

#### 5.4.3.8 bool Billing::isBilling ( ) const

[Billing::isBilling](#). Return if it's a billing or a quote.

## Returns

if it's billing or a quote

#### 5.4.3.9 bool Billing::operator!= ( const Billing & b )

[Billing::operator !=](#) defines the operator "!=" to compare two [Billing](#) and to see if they are different.

## Parameters

<i>b</i>	the <a href="#">Billing</a> to compare with the current <a href="#">Billing</a>
----------	---

## Returns

true if the [Billing](#) are different else false

#### 5.4.3.10 bool Billing::operator== ( const Billing & b )

[Billing::operator ==](#) define the operator "==" to compare two billings and to see if they are the same.

## Parameters

<i>b</i>	the <a href="#">Billing</a> to compare with the current <a href="#">Billing</a>
----------	---

## Returns

true if they are the same billings else false

#### 5.4.3.11 void Billing::setDate ( const QDate & getDate )

[Billing::setDate](#). Modify *date* of the [Billing](#)

## Parameters

<i>getDate</i>	the new date of the <a href="#">Billing</a>
----------------	---

#### 5.4.3.12 void Billing::setDescription ( const QString & getDescription )

[Billing::setDescription](#). Modify the description of [Billing](#)

## Parameters

---



<i>getDescription</i>	Modify the description with <i>getDescription</i>
-----------------------	---

5.4.3.13 void Billing::setIsBilling ( bool *isBilling* )

[Billing::setIsBilling](#). Modify *isBilling* of [Billing](#).

Parameters

<i>isBilling</i>	
------------------	--

5.4.3.14 void Billing::setNumber ( int *getNumber* )

[Billing::setNumber](#). Modify *\_number* of [Billing](#).

Parameters

<i>getNumber</i>	the new number of the <a href="#">Billing</a>
------------------	---

5.4.3.15 void Billing::setTitle ( const QString & *getTitle* )

[Billing::setTitle](#). Modify the title of [Billing](#)

Parameters

<i>getTitle</i>	Modify the title with <i>getTitle</i>
-----------------	---------------------------------------

The documentation for this class was generated from the following files:

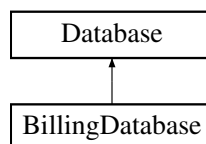
- src/models/billing.h
- src/models/billing.cpp

## 5.5 BillingDatabase Class Reference

The [BillingDatabase](#) class [Billing](#) (or Quote) table database.

```
#include <billingdatabase.h>
```

Inheritance diagram for BillingDatabase:



### Public Member Functions

- [Billing](#) \* [getBilling](#) (const int pld)  
*BillingDatabase::getCustomer* get informations about the billing identified by pld
- QStandardItemModel \* [getBillingsTable](#) (const int idProject) throw (DbException\*)  
*BillingDatabase::getBillingsTable* Return an item model of billings for QTableView.
- int [addBilling](#) (const [Billing](#) &)  
*BillingDatabase::addBilling* Add the billing pBilling to the database.

- void [updateBilling](#) (const [Billing](#) &)  
*BillingDatabase::updateCustomer* Update informations about the billing *pCustomer*
- void [removeBilling](#) (const int *pld*)  
*BillingDatabase::removeCustomer* Remove the billing with the id *pld*
- void [addBillingProject](#) (const int *idProject*, const int *idBilling*, const int *idContributory*)  
*BillingDatabase::addBillingProject* Link a project, a billing and a contributory in the table *BillingProject*.
- int [getMaxBillingNumber](#) ()
- int [getMaxQuoteNuber](#) ()

## Static Public Member Functions

- static [BillingDatabase](#) \* [instance](#) () throw (DbException\*)  
*BillingDatabase::getInstance* Return an instance of [BillingDatabase](#)

## Additional Inherited Members

### 5.5.1 Detailed Description

The [BillingDatabase](#) class [Billing](#) (or Quote) table database.

#### Author

Cédric Rohaut

#### See also

[Database](#)  
[Billing/Quote](#)

### 5.5.2 Member Function Documentation

#### 5.5.2.1 int BillingDatabase::addBilling ( const [Billing](#) & *pBilling* )

[BillingDatabase::addBilling](#) Add the billing *pBilling* to the database.

#### Returns

billing id

#### 5.5.2.2 void BillingDatabase::addBillingProject ( const int *idProject*, const int *idBilling*, const int *idContributory* )

[BillingDatabase::addBillingProject](#) Link a project, a billing and a contributory in the table *BillingProject*.

#### Parameters

<i>idProject</i>	<a href="#">Project</a> id
<i>idBilling</i>	<a href="#">Billing</a> id
<i>idContributory</i>	<a href="#">Contributory</a> id

#### 5.5.2.3 [Billing](#) \* BillingDatabase::getBilling ( const int *pld* )

[BillingDatabase::getCustomer](#) get informations about the billing identified by *pld*

## Parameters

<i>pld</i>	billing id
------------	------------

## Returns

the [Billing](#)

5.5.2.4 `QStandardItemModel * BillingDatabase::getBillingsTable ( const int idProject ) throw DbException *`

[BillingDatabase::getBillingsTable](#) Return an item model of billings for QTableView.

## Parameters

<i>pld</i>	the project id of the billings returned
------------	---

## Exceptions

<a href="#">DbException</a>	
-----------------------------	--

## Returns

QStandardItemModel an item model for QTableView

5.5.2.5 `BillingDatabase * BillingDatabase::instance ( ) throw DbException *` `[static]`

[BillingDatabase::getInstance](#) Return an instance of [BillingDatabase](#)

## See also

[DbException](#)

## Returns

Instance of [BillingDatabase](#)

5.5.2.6 `void BillingDatabase::removeBilling ( const int pld )`

[BillingDatabase::removeCustomer](#) Remove the billing with the id *pld*

## Parameters

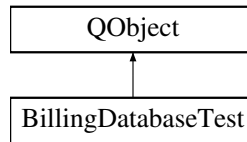
<i>pld</i>	billing id
------------	------------

The documentation for this class was generated from the following files:

- `src/database/billingdatabase.h`
- `src/database/billingdatabase.cpp`

## 5.6 BillingDatabaseTest Class Reference

Inheritance diagram for BillingDatabaseTest:

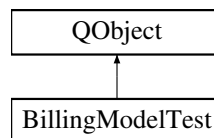


The documentation for this class was generated from the following files:

- tests/database/billingdatabasetest.h
- tests/database/billingdatabasetest.cpp

## 5.7 BillingModelTest Class Reference

Inheritance diagram for BillingModelTest:



The documentation for this class was generated from the following files:

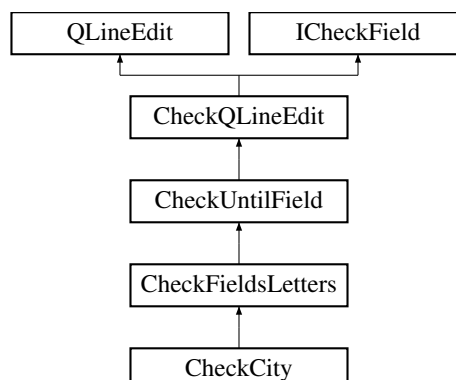
- tests/models/billingmodeltest.h
- tests/models/billingmodeltest.cpp

## 5.8 CheckCity Class Reference

The [CheckCity](#) class Line Edit of City with a check icon.

```
#include <checkcity.h>
```

Inheritance diagram for CheckCity:



### Public Member Functions

- [CheckCity](#) (QWidget \*w=0, QPushButton \*btn=0)  
*CheckCity::CheckCity* Construct a *CheckCity*.

## Additional Inherited Members

### 5.8.1 Detailed Description

The [CheckCity](#) class Line Edit of City with a check icon.

### 5.8.2 Constructor & Destructor Documentation

#### 5.8.2.1 [CheckCity::CheckCity](#) ( [QWidget](#) \* *w* = 0, [QPushButton](#) \* *btn* = 0 )

[CheckCity::CheckCity](#) Construct a [CheckCity](#).

Parameters

<i>w</i>	<a href="#">QWidget</a> linked to <a href="#">CheckCity</a>
----------	---

The documentation for this class was generated from the following files:

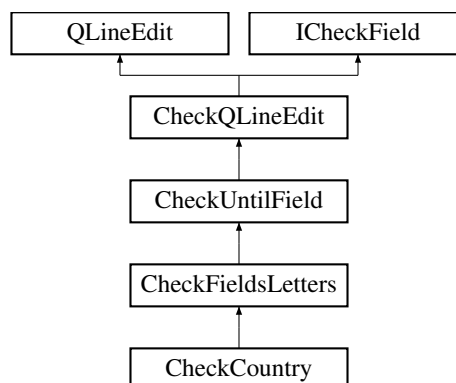
- `src/widgets/checkfields/checkcity.h`
- `src/widgets/checkfields/checkcity.cpp`

## 5.9 CheckCountry Class Reference

[CheckCountry::CheckCountry](#) Line Edit of country with a check icon.

```
#include <checkcountry.h>
```

Inheritance diagram for [CheckCountry](#):



## Public Member Functions

- [CheckCountry](#) ([QWidget](#) \**w*=0, [QPushButton](#) \**btn*=0)  
*CheckEmail::CheckCountry* Construct a [CheckCountry](#).

## Additional Inherited Members

### 5.9.1 Detailed Description

[CheckCountry::CheckCountry](#) Line Edit of country with a check icon.

## 5.9.2 Constructor & Destructor Documentation

### 5.9.2.1 CheckCountry::CheckCountry ( QWidget \* *w* = 0, QPushButton \* *btn* = 0 )

CheckEmail::CheckCountry Construct a [CheckCountry](#).

## Parameters

<i>w</i>	QWidget linked to <a href="#">CheckCountry</a>
----------	--

The documentation for this class was generated from the following files:

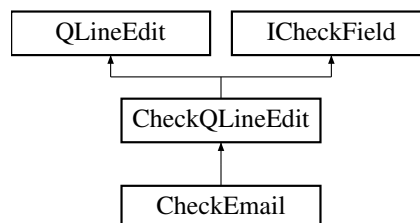
- src/widgets/checkfields/checkcountry.h
- src/widgets/checkfields/checkcountry.cpp

## 5.10 CheckEmail Class Reference

The [CheckEmail](#) class Line Edit of email with a check icon.

```
#include <checkemail.h>
```

Inheritance diagram for CheckEmail:



### Public Member Functions

- [CheckEmail](#) (QWidget \*w=0, QPushButton \*btn=0)  
[CheckEmail::CheckEmail](#) Construct a CheckMail.
- bool [check](#) (const QString text)  
[CheckEmail::check](#) Check if the field email is valid. To be valid, an email address should be under this form↔  
: me@me . xx An email address need:

### Additional Inherited Members

#### 5.10.1 Detailed Description

The [CheckEmail](#) class Line Edit of email with a check icon.

#### 5.10.2 Constructor & Destructor Documentation

##### 5.10.2.1 CheckEmail::CheckEmail ( QWidget \* w = 0 , QPushButton \* btn = 0 )

[CheckEmail::CheckEmail](#) Construct a CheckMail.

## Parameters

<i>w</i>	QWidget linked to <a href="#">CheckEmail</a>
----------	--

#### 5.10.3 Member Function Documentation

#### 5.10.3.1 bool CheckEmail::check ( const QString text ) [virtual]

**CheckEmail::check** Check if the field email is valid. To be valid, an email address should be under this form: `me@me.xx` An email address need:

- 1 character [A-Z] or [a-z] minimum before the character `</i>`
- the character '@'
- 1 character [A-Z] or [a-z] after the character `</i>` minimum and before the character `.`
- 1 character [A-Z] or [a-z] minimum after the character `.` Return TRUE if email address is valid, else FALSE

##### Parameters

text	
------	--

##### Returns

boolean

Implements [ICheckField](#).

The documentation for this class was generated from the following files:

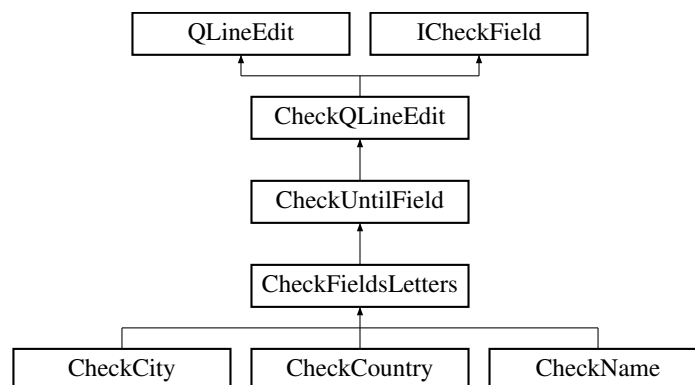
- src/widgets/checkfields/checkemail.h
- src/widgets/checkfields/checkemail.cpp

## 5.11 CheckFieldsLetters Class Reference

The [CheckFieldsLetters](#) class Field with only letters (no numbers)

```
#include <checkfieldsletters.h>
```

Inheritance diagram for CheckFieldsLetters:



### Public Member Functions

- [CheckFieldsLetters](#) (QWidget \*w=0, QPushButton \*btn=0)  
[CheckFieldsLetters::CheckFieldsLetters](#) Construct a [CheckFieldsLetters](#).
- bool [check](#) (QString text)  
[CheckName::check](#) Check if the field contains only letters.



## Additional Inherited Members

### 5.11.1 Detailed Description

The [CheckFieldsLetters](#) class Field with only letters (no numbers)

### 5.11.2 Constructor & Destructor Documentation

#### 5.11.2.1 `CheckFieldsLetters::CheckFieldsLetters ( QWidget * w = 0, QPushButton * btn = 0 )`

[CheckFieldsLetters::CheckFieldsLetters](#) Construct a [CheckFieldsLetters](#).

Parameters

<i>w</i>	QWidget linked to <a href="#">CheckFieldsLetters</a>
----------	--

### 5.11.3 Member Function Documentation

#### 5.11.3.1 `bool CheckFieldsLetters::check ( QString text ) [virtual]`

[CheckName::check](#) Check if the field contains only letters.

Parameters

<i>text</i>	
-------------	--

Returns

boolean

Implements [ICheckField](#).

The documentation for this class was generated from the following files:

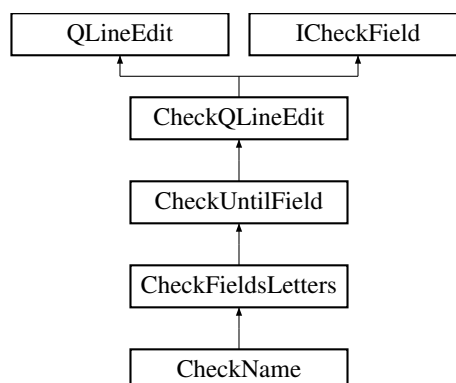
- `src/widgets/checkfields/checkfieldsletters.h`
- `src/widgets/checkfields/checkfieldsletters.cpp`

## 5.12 CheckName Class Reference

The [CheckName](#) class Line edit of name with a check icon.

```
#include <checkname.h>
```

Inheritance diagram for CheckName:



## Public Member Functions

- [CheckName](#) (QWidget \*w=0, QPushButton \*btn=0)  
[CheckName::CheckName](#) Construct a [CheckName](#).

## Additional Inherited Members

### 5.12.1 Detailed Description

The [CheckName](#) class Line edit of name with a check icon.

### 5.12.2 Constructor & Destructor Documentation

#### 5.12.2.1 CheckName::CheckName ( QWidget \* w = 0, QPushButton \* btn = 0 )

[CheckName::CheckName](#) Construct a [CheckName](#).

Parameters

<i>w</i>	QWidget linked to <a href="#">CheckName</a>
----------	---

The documentation for this class was generated from the following files:

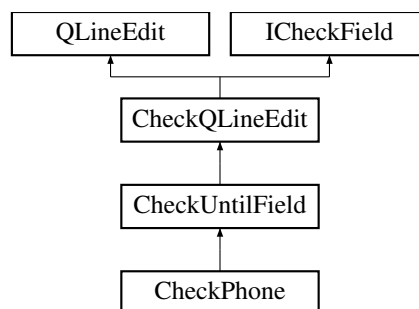
- src/widgets/checkfields/checkname.h
- src/widgets/checkfields/checkname.cpp

## 5.13 CheckPhone Class Reference

The [CheckPhone](#) class Line Edit of Phone number with a check icon.

```
#include <checkphone.h>
```

Inheritance diagram for CheckPhone:



## Public Member Functions

- [CheckPhone](#) (QWidget \*w=0, QPushButton \*btn=0)  
[CheckPhone::CheckPhone](#) Construct a [CheckPhone](#).
- bool [check](#) (QString text)  
[CheckPhone::check](#) Check if the field is valid. To be valid, a name should be composed of a character.
- QString [getCountry](#) () const  
[CheckPhone::getCountry](#) Return the country linked to current field.
- void [setCountry](#) (const QString &country)  
[CheckPhone::setCountry](#) Modify the country linked to field.

## Additional Inherited Members

### 5.13.1 Detailed Description

The [CheckPhone](#) class Line Edit of Phone number with a check icon.

### 5.13.2 Constructor & Destructor Documentation

#### 5.13.2.1 `CheckPhone::CheckPhone ( QWidget * w = 0, QPushButton * btn = 0 )`

[CheckPhone::CheckPhone](#) Construct a [CheckPhone](#).

Parameters

<i>w</i>	QWidget linked to <a href="#">CheckPhone</a>
----------	--

### 5.13.3 Member Function Documentation

#### 5.13.3.1 `bool CheckPhone::check ( QString text ) [virtual]`

[CheckPhone::check](#) Check if the field is valid. To be valid, a name should be composed of a character.

Parameters

<i>text</i>	
-------------	--

Returns

boolean

Implements [ICheckField](#).

#### 5.13.3.2 `QString CheckPhone::getCountry ( ) const`

[CheckPhone::getCountry](#) Return the country linked to current field.

Returns

#### 5.13.3.3 `void CheckPhone::setCountry ( const QString & country )`

[CheckPhone::setCountry](#) Modify the *country* linked to field.

Parameters

<i>country</i>	New country
----------------	-------------

The documentation for this class was generated from the following files:

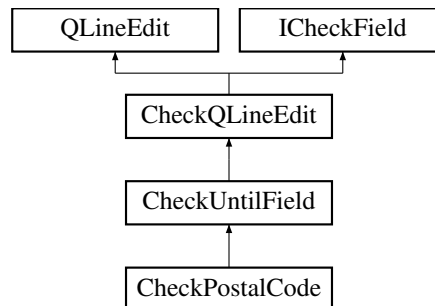
- `src/widgets/checkfields/checkphone.h`
- `src/widgets/checkfields/checkphone.cpp`

## 5.14 CheckPostalCode Class Reference

The [CheckPostalCode](#) class Line Edit of postal code with a check icon.

```
#include <checkpostalcode.h>
```

Inheritance diagram for CheckPostalCode:



### Public Member Functions

- [CheckPostalCode](#) (QWidget \*w=0, QPushButton \*btn=0)  
*[CheckPostalCode::CheckPostalCode](#) Construct a [CheckPostalCode](#).*
- bool [check](#) (QString text)  
*[CheckPostalCode::check](#) Check if the field is valid. To be valid, a name should be composed of a character.*
- QString [getCountry](#) () const  
*[CheckPostalCode::getCountry](#) Return the country linked to current field.*
- void [setCountry](#) (const QString &country)  
*[CheckPostalCode::setCountry](#) Modify the country linked to field.*

### Additional Inherited Members

#### 5.14.1 Detailed Description

The [CheckPostalCode](#) class Line Edit of postal code with a check icon.

#### 5.14.2 Constructor & Destructor Documentation

##### 5.14.2.1 [CheckPostalCode::CheckPostalCode](#) ( QWidget \* w = 0, QPushButton \* btn = 0 )

[CheckPostalCode::CheckPostalCode](#) Construct a [CheckPostalCode](#).

Parameters

<i>w</i>	QWidget linked to <a href="#">CheckPostalCode</a>
----------	---

#### 5.14.3 Member Function Documentation

##### 5.14.3.1 bool [CheckPostalCode::check](#) ( QString *text* ) [virtual]

[CheckPostalCode::check](#) Check if the field is valid. To be valid, a name should be composed of a character.

## Parameters

<i>text</i>	
-------------	--

## Returns

boolean

Implements [ICheckField](#).

## 5.14.3.2 QString CheckPostalCode::getCountry ( ) const

[CheckPostalCode::getCountry](#) Return the country linked to current field.

## Returns

country

## 5.14.3.3 void CheckPostalCode::setCountry ( const QString &amp; country )

[CheckPostalCode::setCountry](#) Modify the *country* linked to field.

## Parameters

<i>country</i>	New country
----------------	-------------

The documentation for this class was generated from the following files:

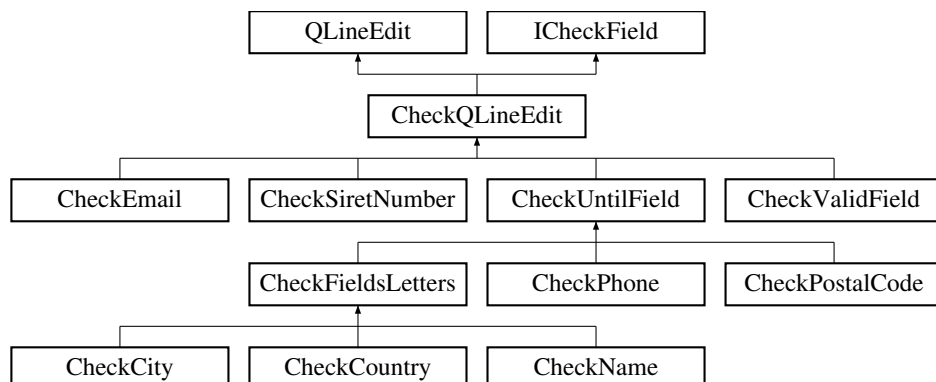
- src/widgets/checkfields/checkpostalcode.h
- src/widgets/checkfields/checkpostalcode.cpp

## 5.15 CheckQLineEdit Class Reference

The [CheckQLineEdit](#) class QLineEdit custom with a check of text inputed.

```
#include <checkqlineedit.h>
```

Inheritance diagram for CheckQLineEdit:



## Public Slots

- void [fieldTextChanged](#) (const QString &text)  
[CheckQLineEdit::fieldTextChanged](#) For each new characater inputed or removed, displays an icon to show if the field is valid or not.

## Public Member Functions

- [CheckQLineEdit](#) (QWidget \*parent=0, QPushButton \*btn=0)  
*CheckQLineEdit::CheckQLineEdit Construct a [CheckQLineEdit](#).*
- void [displayCheckValidFieldIcon](#) ()  
*CheckQLineEdit::displayCheckValidFieldIcon Display a valid icon into the field.*
- void [displayCheckNoValidFieldIcon](#) ()  
*CheckQLineEdit::displayCheckNoValidFieldIcon Display a "no valid" icon into the field.*
- QPushButton \* [getBtnValid](#) () const  
*CheckQLineEdit::getBtnValid.*
- void [setBtnValid](#) (QPushButton \*getBtnValid)  
*CheckQLineEdit::setBtnValid.*
- bool [isValid](#) ()  
*isValid Return true if the current field if valid*

### 5.15.1 Detailed Description

The [CheckQLineEdit](#) class QLineEdit custom with a check of text inputed.

### 5.15.2 Constructor & Destructor Documentation

#### 5.15.2.1 CheckQLineEdit::CheckQLineEdit ( QWidget \* parent = 0, QPushButton \* btn = 0 ) [explicit]

[CheckQLineEdit::CheckQLineEdit](#) Construct a [CheckQLineEdit](#).

Parameters

<i>parent</i>	
---------------	--

### 5.15.3 Member Function Documentation

#### 5.15.3.1 QPushButton \* CheckQLineEdit::getBtnValid ( ) const

[CheckQLineEdit::getBtnValid.](#)

Returns

a

#### 5.15.3.2 bool CheckQLineEdit::isValid ( )

*isValid Return true if the current field if valid*

Returns

boolean

#### 5.15.3.3 void CheckQLineEdit::setBtnValid ( QPushButton \* getBtnValid )

[CheckQLineEdit::setBtnValid.](#)

## Parameters

<code>getBtnValid</code>	
--------------------------	--

The documentation for this class was generated from the following files:

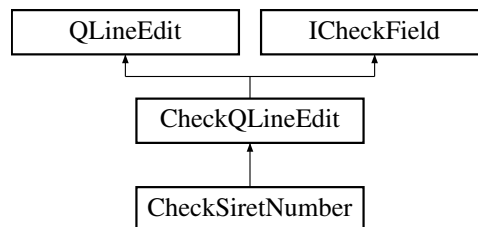
- `src/widgets/checkfields/checkqlineedit.h`
- `src/widgets/checkfields/checkqlineedit.cpp`

## 5.16 CheckSiretNumber Class Reference

The [CheckSiretNumber](#) class Line Edit with a check icon.

```
#include <checksiretnumber.h>
```

Inheritance diagram for CheckSiretNumber:



### Public Member Functions

- [CheckSiretNumber](#) (QWidget \*w=0, QPushButton \*btn=0)  
[CheckSiretNumber::CheckSiretNumber](#) Construct a [CheckSiretNumber](#).
- bool [check](#) (QString text)  
[CheckSiretNumber::check](#) Check if the field noSiret is valid. To be valid, a SIRET number should be composed of numbers.

### Additional Inherited Members

#### 5.16.1 Detailed Description

The [CheckSiretNumber](#) class Line Edit with a check icon.

#### 5.16.2 Constructor & Destructor Documentation

##### 5.16.2.1 CheckSiretNumber::CheckSiretNumber ( QWidget \* w = 0, QPushButton \* btn = 0 )

[CheckSiretNumber::CheckSiretNumber](#) Construct a [CheckSiretNumber](#).

## Parameters

<code>w</code>	QWidget linked to <a href="#">CheckSiretNumber</a>
----------------	--

#### 5.16.3 Member Function Documentation

5.16.3.1 `bool CheckSiretNumber::check ( QString text )` `[virtual]`

[CheckSiretNumber::check](#) Check if the field noSiret is valid. To be valid, a SIRET number should be composed of numbers.



## Parameters

<i>text</i>	
-------------	--

## Returns

boolean

Implements [ICheckField](#).

The documentation for this class was generated from the following files:

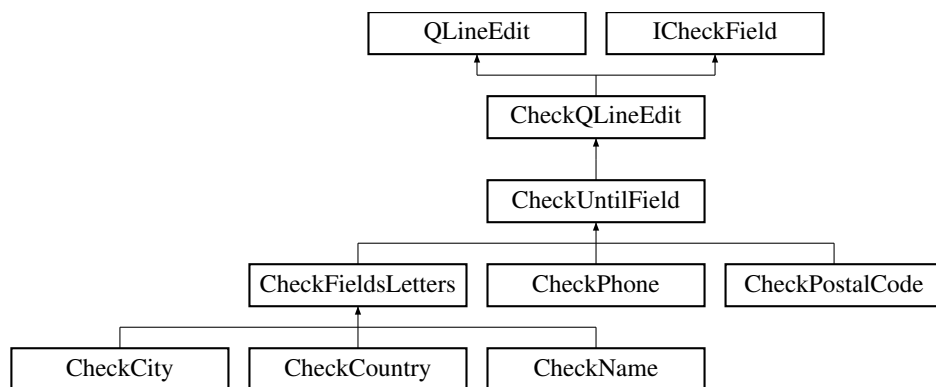
- src/widgets/checkfields/checksiretnumber.h
- src/widgets/checkfields/checksiretnumber.cpp

## 5.17 CheckUntilField Class Reference

The [CheckUntilField](#) class.

```
#include <checkuntilfield.h>
```

Inheritance diagram for CheckUntilField:



### Public Member Functions

- [CheckUntilField](#) (QWidget \*w=0, QPushButton \*btn=0)  
[CheckUntilField::CheckUntilField](#) Construct a [CheckUntilField](#).
- bool [check](#) (QString text)  
[CheckUntilField::check](#) Check if the field is valid. To be valid, a name should be composed of a character.

### Additional Inherited Members

#### 5.17.1 Detailed Description

The [CheckUntilField](#) class.

#### 5.17.2 Constructor & Destructor Documentation

##### 5.17.2.1 CheckUntilField::CheckUntilField ( QWidget \* w = 0, QPushButton \* btn = 0 )

[CheckUntilField::CheckUntilField](#) Construct a [CheckUntilField](#).

## Parameters

<i>w</i>	QWidget linked to <a href="#">CheckUntilField</a>
----------	---

## 5.17.3 Member Function Documentation

5.17.3.1 `bool CheckUntilField::check ( QString text ) [virtual]`

[CheckUntilField::check](#) Check if the field is valid. To be valid, a name should be composed of a character.

## Parameters

<i>text</i>	
-------------	--

## Returns

boolean

Implements [ICheckField](#).

The documentation for this class was generated from the following files:

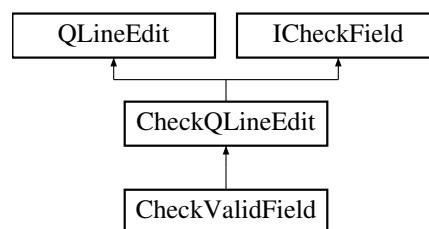
- `src/widgets/checkfields/checkuntilfield.h`
- `src/widgets/checkfields/checkuntilfield.cpp`

## 5.18 CheckValidField Class Reference

The [CheckValidField](#) class Check field not required.

```
#include <checkvalidfield.h>
```

Inheritance diagram for CheckValidField:



## Public Member Functions

- [CheckValidField](#) (QWidget \*w=0, QPushButton \*btn=0)  
[CheckValidField::CheckValidField](#).
- `bool check` (QString text)  
[CheckValidField::check](#) Return TRUE : the field is not required.

## Additional Inherited Members

## 5.18.1 Detailed Description

The [CheckValidField](#) class Check field not required.

## 5.18.2 Constructor & Destructor Documentation

### 5.18.2.1 CheckValidField::CheckValidField ( QWidget \* *w* = 0, QPushButton \* *btn* = 0 )

[CheckValidField::CheckValidField](#).

Parameters

<i>w</i>	QWidget linked to <a href="#">CheckValidField</a>
----------	---

## 5.18.3 Member Function Documentation

### 5.18.3.1 bool CheckValidField::check ( QString *text* ) [virtual]

[CheckValidField::check](#) Return TRUE : the field is not required.

Parameters

<i>text</i>	
-------------	--

Returns

boolean

Implements [ICheckField](#).

The documentation for this class was generated from the following files:

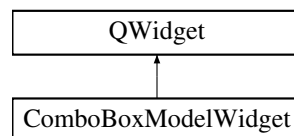
- src/widgets/checkfields/checkvalidfield.h
- src/widgets/checkfields/checkvalidfield.cpp

## 5.19 ComboBoxModelWidget Class Reference

The [ComboBoxModelWidget](#) class TODO.

```
#include <comboboxmodelwidget.h>
```

Inheritance diagram for ComboBoxModelWidget:



## Public Member Functions

- **ComboBoxModelWidget** (QWidget \*parent=0)

## 5.19.1 Detailed Description

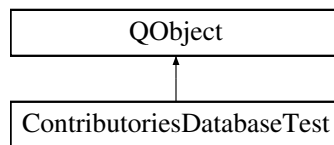
The [ComboBoxModelWidget](#) class TODO.

The documentation for this class was generated from the following files:

- src/widgets/comboboxmodelwidget.h
- src/widgets/comboboxmodelwidget.cpp

## 5.20 ContributoriesDatabaseTest Class Reference

Inheritance diagram for ContributoriesDatabaseTest:



The documentation for this class was generated from the following files:

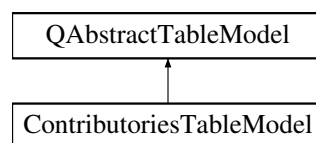
- tests/database/contributoriesdatabasetest.h
- tests/database/contributoriesdatabasetest.cpp

## 5.21 ContributoriesTableModel Class Reference

The [ContributoriesTableModel](#) class for a custom table for contributories widget.

```
#include <contributoriesTableModel.h>
```

Inheritance diagram for ContributoriesTableModel:



### Public Member Functions

- [ContributoriesTableModel](#) (QObject \*parent=0)  
*ContributoriesTableModel Construct a ContributoriesTableModel.*
- int [rowCount](#) (const QModelIndex &) const  
*rowCount Number of contributories row*
- int [columnCount](#) (const QModelIndex &) const  
*columnCount Number of column of a contributory*
- QVariant [data](#) (const QModelIndex &index, int role) const  
*data Obtains data of a specify cell*
- QVariant [headerData](#) (int section, Qt::Orientation orientation, int role) const  
*headerData Obtains header title of table*
- bool [setData](#) (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole)  
*setData Change data of a cell*
- void [append](#) (const [Contributory](#) &contributory)  
*append Add a new line in table*
- void [remove](#) (const int i)  
*remove Remove a line*
- Qt::ItemFlags [flags](#) (const QModelIndex &index) const  
*flags Differents table flags*
- QList< [Contributory](#) > [getContributories](#) ()  
*getContributories Get all contributories of table*
- int [count](#) ()  
*count Number of contributories in table*

### 5.21.1 Detailed Description

The [ContributoriesTableModel](#) class for a custom table for contributories widget.

#### Author

Antoine de Roquemaurel

#### See also

[Contributory](#)

### 5.21.2 Constructor & Destructor Documentation

#### 5.21.2.1 ContributoriesTableModel::ContributoriesTableModel ( *QObject* \* *parent* = 0 )

[ContributoriesTableModel](#) Construct a [ContributoriesTableModel](#).

#### Parameters

<i>parent</i>	Parent widget
---------------	---------------

### 5.21.3 Member Function Documentation

#### 5.21.3.1 void ContributoriesTableModel::append ( const [Contributory](#) & *contributory* )

append Add a new line in table

#### Parameters

<i>contributory</i>	The new contributory
---------------------	----------------------

#### 5.21.3.2 int ContributoriesTableModel::columnCount ( const [QModelIndex](#) & ) const

columnCount Number of column of a contributory

#### Returns

The number of column

#### 5.21.3.3 int ContributoriesTableModel::count ( )

count Number of contributories in table

#### Returns

The number of contributories

#### 5.21.3.4 [QVariant](#) ContributoriesTableModel::data ( const [QModelIndex](#) & *index*, int *role* ) const

data Obtains data of a specify cell

## Parameters

<i>index</i>	The cell who we want data
<i>role</i>	The role of set

## Returns

The data of cell

**5.21.3.5 Qt::ItemFlags ContributoriesTableModel::flags ( const QModelIndex & *index* ) const**

flags Differents table flags

## Parameters

<i>index</i>	The cell who we want to know flags
--------------	------------------------------------

## Returns

Flags

**5.21.3.6 QList< Contributory > ContributoriesTableModel::getContributories ( )**

getContributories Get all contributories of table

## Returns

The contributory list

**5.21.3.7 QVariant ContributoriesTableModel::headerData ( int *section*, Qt::Orientation *orientation*, int *role* ) const**

headerData Obtains header title of table

## Parameters

<i>section</i>	The number of column
<i>orientation</i>	The table orientation
<i>role</i>	

## Returns

The Title header of column

**5.21.3.8 void ContributoriesTableModel::remove ( const int *i* )**

remove Remove a line

## Parameters

<i>i</i>	The number of line to remove
----------	------------------------------

**5.21.3.9 int ContributoriesTableModel::rowCount ( const QModelIndex & ) const**

rowCount Number of contributories row

**Returns**

The number of contributories

**5.21.3.10** `bool ContributoriesTableModel::setData ( const QModelIndex & index, const QVariant & value, int role = Qt::EditRole )`

setData Change data of a cell

**Parameters**

<i>index</i>	The cell to change data
<i>value</i>	The new value
<i>role</i>	The role of cell

**Returns**

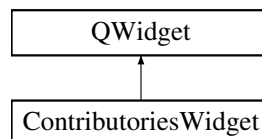
True if we could edit

The documentation for this class was generated from the following files:

- src/widgets/widgetsmodels/contributoriestablemodel.h
- src/widgets/widgetsmodels/contributoriestablemodel.cpp

## 5.22 ContributoriesWidget Class Reference

Inheritance diagram for ContributoriesWidget:

**Public Slots**

- void [add](#) (void)  
*add Add a new empty contributory*
- void [remove](#) (void)  
*remove Remove the current contributory*

**Signals**

- void **contributoryChanged** ()

**Public Member Functions**

- **ContributoriesWidget** (QSharedPointer< [Customer](#) > c, QWidget \*parent=0)
- QList< [Contributory](#) > [getContributories](#) () const  
*getContributories Get contributories List*
- int **count** ()

## 5.22.1 Member Function Documentation

### 5.22.1.1 QList< [Contributory](#) > ContributoriesWidget::getContributories ( ) const

getContributories Get contributories List

Returns

The documentation for this class was generated from the following files:

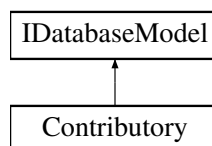
- src/widgets/contributorieswidget.h
- src/widgets/contributorieswidget.cpp

## 5.23 [Contributory](#) Class Reference

The [Contributory](#) class.

```
#include <contributory.h>
```

Inheritance diagram for [Contributory](#):



### Public Member Functions

- [Contributory](#) ()  
*[Contributory::Contributory](#) Construct a [Contributory](#).*
- [Contributory](#) (int id)  
*[Contributory::Contributory](#) Construct a [Contributory](#) and get data in database.*
- void [commit](#) ()  
*[Contributory::commit](#) Update or insert a contributory to the database.*
- void [hydrat](#) (int id)  
*[Contributory::hydrat](#) Get data about the [Contributory](#) which is specified by the identify id*
- void [remove](#) ()  
*[Contributory::remove](#) Remove the current [Contributory](#).*
- [Project](#) \* [getProject](#) () const  
*[Contributory::getProject](#) Return the project linked to this [Contributory](#).*
- void [setProject](#) ([Project](#) \*id)  
*[Contributory::setProject](#) Modify the identify id of the [Project](#) linke to this [Contributory](#).*
- double [getNbHours](#) () const  
*[getNbHours](#) Number of work hour of a contributory*
- void [setNbHours](#) (double value)  
*[setNbHours](#) Change nbHours*
- QString [getDescription](#) () const  
*[getDescription](#) Description of a contributory*
- void [setDescription](#) (const QString &[getDescription](#))  
*[setDescription](#) Change the contributory description*



## Additional Inherited Members

### 5.23.1 Detailed Description

The [Contributory](#) class.

Author

### 5.23.2 Constructor & Destructor Documentation

#### 5.23.2.1 [Contributory::Contributory](#) ( int *id* )

[Contributory::Contributory](#) Construct a [Contributory](#) and get data in database.

Parameters

<i>id</i>	<a href="#">Contributory</a> 's id
-----------	------------------------------------

### 5.23.3 Member Function Documentation

#### 5.23.3.1 [QString](#) [Contributory::getDescription](#) ( ) const

[getDescription](#) Description of a contributory

Returns

The description

#### 5.23.3.2 [double](#) [Contributory::getNbHours](#) ( ) const

[getNbHours](#) Number of work hour of a contributory

Returns

Then number of hours

#### 5.23.3.3 [Project](#) \* [Contributory::getProject](#) ( ) const

[Contributory::getProject](#) Return the project linked to this [Contributory](#).

Returns

[Project](#) linked to this [Contributory](#)

#### 5.23.3.4 [void](#) [Contributory::hydrat](#) ( int *id* ) [virtual]

[Contributory::hydrat](#) Get data about the [Contributory](#) which is specified by the identify *id*

## Parameters

<i>id</i>	<a href="#">Contributory</a> identify
-----------	---------------------------------------

Implements [IDatabaseModel](#).

#### 5.23.3.5 void Contributory::setDescription ( const QString & *getDescription* )

setDescription Change the contributory description

## Parameters

<i>getDescription</i>	The new description
-----------------------	---------------------

#### 5.23.3.6 void Contributory::setNbHours ( double *value* )

setNbHours Change nbHours

## Parameters

<i>value</i>	The new value of nbHours
--------------	--------------------------

#### 5.23.3.7 void Contributory::setProject ( [Project](#) \* *id* )

[Contributory::setProject](#) Modify the identify *id* of the [Project](#) linke to this [Contributory](#).

## Parameters

<i>id</i>	<a href="#">Project</a> Identify
-----------	----------------------------------

The documentation for this class was generated from the following files:

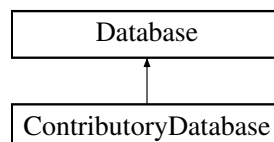
- src/models/contributory.h
- src/models/contributory.cpp

## 5.24 ContributoryDatabase Class Reference

The [ContributoryDatabase](#) class [Contributory](#) (or Quote) table database.

```
#include <contributorydatabase.h>
```

Inheritance diagram for ContributoryDatabase:



### Public Member Functions

- [Contributory](#) \* [getContributory](#) (const int idContributory)  
*ContributoryDatabase::getCustomer* get informations about the [Contributory](#) identified by *pld*
- QMap< [Project](#) \*, QList  
 < [Contributory](#) > > [getContributoriesByBilling](#) (const int idBilling)

- int [addContributory](#) (const [Contributory](#) &)  
*ContributoryDatabase::addContributory* Add the [Contributory](#) *pContributory* to the database.
- void [updateContributory](#) (const [Contributory](#) &)  
*ContributoryDatabase::updateCustomer* Update informations about the [Contributory](#) *pCustomer*
- void [removeContributory](#) (const int *pld*)  
*ContributoryDatabase::removeCustomer* Remove the [Contributory](#) with the id *pld*
- [Contributory](#) \* [getContributory](#) (QString &q)  
*getContributory* Obtain a contributory without new query

### Static Public Member Functions

- static [ContributoryDatabase](#) \* [instance](#) () throw (DbException\*)  
*ContributoryDatabase::getInstance* Return an instance of [ContributoryDatabase](#)

### Additional Inherited Members

#### 5.24.1 Detailed Description

The [ContributoryDatabase](#) class [Contributory](#) (or Quote) table database.

#### Author

Cédric Rohaut

#### See also

[Database](#)  
[Contributory/Quote](#)

#### 5.24.2 Member Function Documentation

##### 5.24.2.1 int ContributoryDatabase::addContributory ( const Contributory & pContributory )

[ContributoryDatabase::addContributory](#) Add the [Contributory](#) *pContributory* to the database.

#### Returns

[Contributory](#) id

##### 5.24.2.2 Contributory \* ContributoryDatabase::getContributory ( const int idContributory )

[ContributoryDatabase::getCustomer](#) get informations about the [Contributory](#) identified by *pld*

#### Parameters

<i>idContributory</i>	<a href="#">Contributory</a> id
-----------------------	---------------------------------

#### Returns

the [Contributory](#)

##### 5.24.2.3 Contributory \* ContributoryDatabase::getContributory ( QSqlQuery & q )

[getContributory](#) Obtain a contributory without new query

## Parameters

<i>q</i>	The query to use
----------	------------------

## Returns

The contributory linked to *q*

#### 5.24.2.4 **ContributoryDatabase \* ContributoryDatabase::instance ( ) throw DbException \*** [static]

ContributoryDatabase::getInstance Return an instance of [ContributoryDatabase](#)

## See also

[DbException](#)

## Returns

Instance of [ContributoryDatabase](#)

#### 5.24.2.5 **void ContributoryDatabase::removeContributory ( const int *pld* )**

ContributoryDatabase::removeCustomer Remove the [Contributory](#) with the id *pld*

## Parameters

<i>pld</i>	<a href="#">Contributory</a> id
------------	---------------------------------

The documentation for this class was generated from the following files:

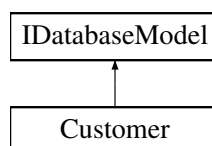
- src/database/contributorydatabase.h
- src/database/contributorydatabase.cpp

## 5.25 Customer Class Reference

The [Customer](#) class [Customer](#).

```
#include <customer.h>
```

Inheritance diagram for Customer:



## Public Member Functions

- [Customer](#) ()  
*Customer::Customer* Construct a [Customer](#).
- [Customer](#) (int id)  
*Customer::Customer* Construct a [Customer](#) who is specified by id
- void [commit](#) ()

- Customer::commit* Update customer data on the database.

  - void **hydrat** (int id)

*Customer::hydrat* Insert into database informations related to the *Customer* who is specified by id
- void **remove** ()

*Customer::remove* Remove the current customer.
- QString **getFirstnameReferent** () const

*Customer::getFirstnameReferent* Return the firstname of the referent *Customer*.
- void **setFirstnameReferent** (const QString &firstnameReferent)

*Customer::setFirstnameReferent* Replace the referent firstname of the customer by firstnameReferent
- QString **getLastnameReferent** () const

*Customer::getLastnameReferent* Return the lastname of the referent *Customer*.
- void **setLastnameReferent** (const QString &lastnameReferent)

*Customer::setLastnameReferent* Replace the referent lastname of the customer by lastnameReferent
- QString **getCompany** () const

*Customer::getCompany* Return the name of the company customer.
- void **setCompany** (const QString &company)

*Customer::setCompany* Replace the company name by company
- QString **getAddress** () const

*Customer::getAddress* Return the address (name and street number) of the customer company.
- void **setAddress** (const QString &address)

*Customer::setAddress* Replace the current company address by the new address
- QString **getPostalCode** () const

*Customer::getPostalCode* Return the postal code of the customer company.
- void **setPostalCode** (const QString &postalCode)

*Customer::setPostalCode* Replace the current postal code of the customer company by the new postalCode
- QString **getCity** () const

*Customer::getCity* Return the city of the customer company.
- void **setCity** (const QString &city)

*Customer::setCity* Replace the current city of the customer company by the new city
- QString **getCountry** () const

*Customer::getCountry* Return the country of the customer company.
- void **setCountry** (const QString &country)

*Customer::setCountry* Replace the country of the *Customer* company by country
- QString **getEmail** () const

*Customer::getEmail* Return the email of the customer company.
- void **setEmail** (const QString &email)

*Customer::setEmail* Replace the email of the *Customer* company by email
- QString **getMobilePhone** () const

*Customer::getMobilePhone* Return the number of the mobile phone of the customer company.
- void **setMobilePhone** (const QString &mobilePhone)

*Customer::setMobilePhone* Replace the current mobile phone of the customer company by mobilePhone
- QString **getPhone** () const

*Customer::getPhone* Return the number of desktop phone.
- void **setPhone** (const QString &phone)

*Customer::setPhone* Replace the current number of desktop phone by phone
- QString **getFax** () const

*Customer::getFax* Return the fax number.
- void **setFax** (const QString &fax)

*Customer::setFax* Replace the current fax number by fax
- bool **operator==** (const *Customer* &c)

*Customer::operator ==* Re-define the operator "==" to compare if the current customer is the same to the other **Customer** c Return TRUE if both customers are the same, else FALSE.

- bool **operator!=** (const **Customer** &c)

*Customer::operator !=* Re-define the operator "!=" to compare if the current customer is different to the other **Customer** c Return TRUE if both customers are different, else FALSE.

## Additional Inherited Members

### 5.25.1 Detailed Description

The **Customer** class **Customer**.

#### Author

Antoine de Roquemaurel  
Florent Berbie

### 5.25.2 Constructor & Destructor Documentation

#### 5.25.2.1 Customer::Customer ( int id )

**Customer::Customer** Construct a **Customer** who is specified by *id*

#### Parameters

<i>id</i>	<b>Customer</b> identify
-----------	--------------------------

### 5.25.3 Member Function Documentation

#### 5.25.3.1 QString Customer::getAddress ( ) const

**Customer::getAddress** Return the address (name and street number) of the customer company.

#### Returns

company address

#### 5.25.3.2 QString Customer::getCity ( ) const

**Customer::getCity** Return the city of the customer company.

#### Returns

city of the customer company

#### 5.25.3.3 QString Customer::getCompany ( ) const

**Customer::getCompany** Return the name of the company customer.

#### Returns

name of the company customer

#### 5.25.3.4 QString Customer::getCountry ( ) const

[Customer::getCountry](#) Return the country of the customer company.

##### Returns

country of the customer company

#### 5.25.3.5 QString Customer::getEmail ( ) const

[Customer::getEmail](#) Return the email of the customer company.

##### Returns

email of the customer company

#### 5.25.3.6 QString Customer::getFax ( ) const

[Customer::getFax](#) Return the fax number.

##### Returns

fax number

#### 5.25.3.7 QString Customer::getFirstnameReferent ( ) const

[Customer::getFirstnameReferent](#) Return the firstname of the referent [Customer](#).

##### Returns

Referent firstname of the customer

#### 5.25.3.8 QString Customer::getLastnameReferent ( ) const

[Customer::getLastnameReferent](#) Return the lastname of the referent [Customer](#).

##### Returns

Referent lastname of the customer

#### 5.25.3.9 QString Customer::getMobilePhone ( ) const

[Customer::getMobilePhone](#) Return the number of the mobile phone of the customer company.

##### Returns

number of mobile phone

#### 5.25.3.10 QString Customer::getPhone ( ) const

[Customer::getPhone](#) Return the number of desktop phone.

##### Returns

number of desktop phone

#### 5.25.3.11 QString Customer::getPostalCode ( ) const

[Customer::getPostalCode](#) Return the postal code of the customer company.

##### Returns

postal code of the customer company

#### 5.25.3.12 void Customer::hydrat ( int *id* ) [virtual]

[Customer::hydrat](#) Insert into database informations related to the [Customer](#) who is specified by *id*

##### Parameters

<i>id</i>	<a href="#">Customer</a> identify
-----------	-----------------------------------

Implements [IDatabaseModel](#).

#### 5.25.3.13 bool Customer::operator!= ( const Customer & *c* )

[Customer::operator !=](#) Re-define the operator "!=" to compare if the current customer is differnt to the other [Customer](#) *c* Return TRUE if both customers are different, else FALSE.

##### Parameters

<i>c</i>	<a href="#">Customer</a> to compare
----------	-------------------------------------

##### Returns

boolean

#### 5.25.3.14 bool Customer::operator== ( const Customer & *c* )

[Customer::operator ==](#) Re-define the operator "==" to compare if the current customer is the same to the other [Customer](#) *c* Return TRUE if both customers are the same, else FALSE.

##### Parameters

<i>c</i>	<a href="#">Customer</a> to compare
----------	-------------------------------------

##### Returns

boolean

#### 5.25.3.15 void Customer::setAddress ( const QString & *address* )

[Customer::setAddress](#) Replace the current company address by the new *address*

##### Parameters

<i>address</i>	New company address
----------------	---------------------

#### 5.25.3.16 void Customer::setCity ( const QString & *city* )

[Customer::setCity](#) Replace the current city of the customer company by the new *city*



## Parameters

<i>city</i>	New city of the customer company
-------------	----------------------------------

5.25.3.17 void Customer::setCompany ( const QString & *company* )

[Customer::setCompany](#) Replace the company name by *company*

## Parameters

<i>company</i>	New company name
----------------	------------------

5.25.3.18 void Customer::setCountry ( const QString & *country* )

[Customer::setCountry](#) Replace the country of the [Customer](#) company by *country*

## Parameters

<i>country</i>	New country of the customer company
----------------	-------------------------------------

5.25.3.19 void Customer::setEmail ( const QString & *email* )

[Customer::setEmail](#) Replace the email of the [Customer](#) company by *email*

## Parameters

<i>email</i>	New email of the customer company
--------------	-----------------------------------

5.25.3.20 void Customer::setFax ( const QString & *fax* )

[Customer::setFax](#) Replace the current fax number by *fax*

## Parameters

<i>fax</i>	new fax number
------------	----------------

5.25.3.21 void Customer::setFirstnameReferent ( const QString & *firstnameReferent* )

[Customer::setFirstnameReferent](#) Replace the referent firstname of the customer by *firstnameReferent*

## Parameters

<i>firstnameReferent</i>	New referent firstname of the <a href="#">Customer</a>
--------------------------	--

5.25.3.22 void Customer::setLastnameReferent ( const QString & *lastnameReferent* )

[Customer::setLastnameReferent](#) Replace the referent lastname of the customer by *lastnameReferent*

## Parameters

---

<i>lastname</i> ↔ <i>Referent</i>	New referent lastname of the <a href="#">Customer</a>
--------------------------------------	---

5.25.3.23 void [Customer::setMobilePhone](#) ( const QString & *mobilePhone* )

[Customer::setMobilePhone](#) Replace the current mobile phone of the customer company by *mobilePhone*

Parameters

<i>mobilePhone</i>	New number of mobile phone
--------------------	----------------------------

5.25.3.24 void [Customer::setPhone](#) ( const QString & *phone* )

[Customer::setPhone](#) Replace the current number of desktop phone by *phone*

Parameters

<i>phone</i>	New number of desktop phone
--------------	-----------------------------

5.25.3.25 void [Customer::setPostalCode](#) ( const QString & *postalCode* )

[Customer::setPostalCode](#) Replace the current postal code of the customer company by the new *postalCode*

Parameters

<i>postalCode</i>	New postal code of the customer company
-------------------	---

The documentation for this class was generated from the following files:

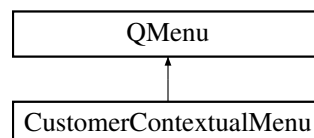
- src/models/customer.h
- src/models/customer.cpp

## 5.26 CustomerContextMenu Class Reference

Display contextual menu on a customer.

```
#include <customercontextualmenu.h>
```

Inheritance diagram for CustomerContextMenu:



### Public Member Functions

- [CustomerContextMenu](#) (QWidget \*w=0)  
*CustomerContextMenu* Construct a new contextual menu.
- [~CustomerContextMenu](#) ()  
*Destruct the contextual menu.*

### 5.26.1 Detailed Description

Display contextual menu on a customer.

#### Author

Antoine de Roquemaurel

### 5.26.2 Constructor & Destructor Documentation

#### 5.26.2.1 CustomerContextMenu::CustomerContextMenu ( QWidget \* w = 0 )

[CustomerContextMenu](#) Construct a new contextual menu.

#### Parameters

<i>w</i>	Parent
----------	--------

The documentation for this class was generated from the following files:

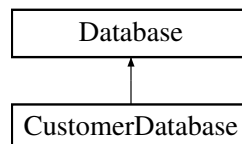
- src/widgets/customercontextualmenu.h
- src/widgets/customercontextualmenu.cpp

## 5.27 CustomerDatabase Class Reference

The [CustomerDatabase](#) class [Customer](#) table database.

```
#include <customerdatabase.h>
```

Inheritance diagram for CustomerDatabase:



### Public Member Functions

- QStandardItemModel \* [getCustomersTable](#) (QString filter="") throw (DbException\*)  
*CustomerDatabase::getCustomersTable* Return an item model of customers for QTableView.
- QStandardItemModel \* [getCustomersTree](#) (QString filter="") throw (DbException\*)  
*CustomerDatabase::getCustomersTree* Return an item model of customers for QTree.
- QSharedPointer< [Customer](#) > [getCustomer](#) (const int pld)  
*CustomerDatabase::getCustomer* get informations about the customer identified by pld
- int [addCustomer](#) (const [Customer](#) &)  
*CustomerDatabase::addCustomer* Add the customer pCustomer to the database.
- void [updateCustomer](#) (const [Customer](#) &)  
*CustomerDatabase::updateCustomer* Update informations about the customer pCustomer
- void [removeCustomer](#) (const int pld)  
*CustomerDatabase::removeCustomer* Remove the customer with the id pld
- int [getNbCustomers](#) ()  
*CustomerDatabase::getNbCustomers* Return the number of customers existing.

## Static Public Member Functions

- static [CustomerDatabase](#) \* [instance](#) () throw (DbException\*)  
*CustomerDatabase::getInstance* Return an instance of [CustomerDatabase](#)

## Additional Inherited Members

### 5.27.1 Detailed Description

The [CustomerDatabase](#) class [Customer](#) table database.

#### Author

Antoine de Roquemaurel

#### See also

[Database](#)  
[Customer](#)

### 5.27.2 Member Function Documentation

#### 5.27.2.1 int CustomerDatabase::addCustomer ( const Customer & pCustomer )

[CustomerDatabase::addCustomer](#) Add the customer *pCustomer* to the database.

#### Returns

customer id

#### 5.27.2.2 QSharedPointer< Customer > CustomerDatabase::getCustomer ( const int pId )

[CustomerDatabase::getCustomer](#) get informations about the customer identified by *pId*

#### Parameters

<i>pId</i>	customer id
------------	-------------

#### Returns

the [Customer](#)

#### 5.27.2.3 QStandardItemModel \* CustomerDatabase::getCustomersTable ( QString filter = " " ) throw DbException \*)

[CustomerDatabase::getCustomersTable](#) Return an item model of customers for QTableView.

#### Author

Manantsoa Razanajatovo

## Parameters

<i>filter</i>	Select only customers who are specified by <i>filter</i>
---------------	--

## Exceptions

<a href="#">DbException</a>	
-----------------------------	--

## Returns

QStandardItemModel an item model for QTableView

#### 5.27.2.4 QStandardItemModel \* CustomerDatabase::getCustomersTree ( QString *filter* = " " ) throw DbException \*)

[CustomerDatabase::getCustomersTree](#) Return an item model of customers for QTree.

## Author

Manantsoa Razanajatovo

## Parameters

<i>filter</i>	Select only customers who are specified by <i>filter</i>
---------------	--

## Exceptions

<a href="#">DbException</a>	
-----------------------------	--

## Returns

QStandardItemModel an item model for QTreeView

#### 5.27.2.5 int CustomerDatabase::getNbCustomers ( )

[CustomerDatabase::getNbCustomers](#) Return the number of customers existing.

## Returns

number of customers

#### 5.27.2.6 CustomerDatabase \* CustomerDatabase::instance ( ) throw DbException \*) [static]

[CustomerDatabase::getInstance](#) Return an instance of [CustomerDatabase](#)

## See also

[DbException](#)

## Returns

Instance of [CustomerDatabase](#)

#### 5.27.2.7 void CustomerDatabase::removeCustomer ( const int *pld* )

[CustomerDatabase::removeCustomer](#) Remove the customer with the id *pld*

## Parameters

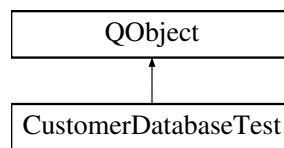
<i>pid</i>	customer id
------------	-------------

The documentation for this class was generated from the following files:

- src/database/customerdatabase.h
- src/database/customerdatabase.cpp

## 5.28 CustomerDatabaseTest Class Reference

Inheritance diagram for CustomerDatabaseTest:



The documentation for this class was generated from the following files:

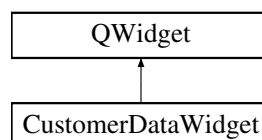
- tests/database/customerdatabasetest.h
- tests/database/customerdatabasetest.cpp

## 5.29 CustomerDataWidget Class Reference

Class for display info of a customer.

```
#include <customerdatawidget.h>
```

Inheritance diagram for CustomerDataWidget:



### Public Member Functions

- **CustomerDataWidget** (QWidget \*parent=0)
- void [printUserData](#) ()  
*printUserData Print Data of current user*
- void [printInformations](#) (int id)  
*printInformations Print Data of customer id*

### 5.29.1 Detailed Description

Class for display info of a customer.

#### Author

Cédric Rohaut

## 5.29.2 Member Function Documentation

### 5.29.2.1 void CustomerDataWidget::printInformations ( int *id* )

printInformations Print Data of customer id

Parameters

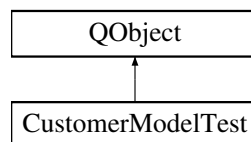
<i>id</i>	of customer to print
-----------	----------------------

The documentation for this class was generated from the following files:

- src/widgets/customerdatawidget.h
- src/widgets/customerdatawidget.cpp

## 5.30 CustomerModelTest Class Reference

Inheritance diagram for CustomerModelTest:



The documentation for this class was generated from the following files:

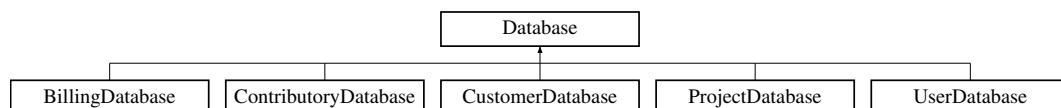
- tests/models/customermodeltest.h
- tests/models/customermodeltest.cpp

## 5.31 Database Class Reference

The **Database** class Master class for all database access.

```
#include <database.h>
```

Inheritance diagram for Database:



### Public Member Functions

- QString **lastError** (const QSqlQuery &q)  
*Database::lastError* Return an error message on the last error occurred during the SQL request q
- void **testCases** ()  
*Database::testCases* Realise a test cases.
- void **cleanDatabase** ()  
*Database::viderDatabase* Clear database.
- void **executeFile** (QString pName)  
*Database::executerFichier* Exeute a specified file named pName

- void `openTransaction` ()  
*Database::openTransaction* Open new transaction.
- void `closeTransaction` ()  
*Database::closeTransaction* Close current transaction.
- void `close` ()  
*Database::close* Close database access.
- void `open` ()  
*Database::open* Open database.
- `~Database` ()  
*Database::~Database* Suppression object, and close database access.
- void `setDatabase` (QSqlDatabase sql)  
*Database::setDatabase* Set database.
- void `createDatabase` ()  
*Database::creerDatabase* Create a new database.
- void `updateBillingNumber` ()

### Static Public Member Functions

- static `Database * instance` () throw (DbException\*)  
*Database::getInstance* Return an instance of *Database*.

### Protected Member Functions

- `Database` () throw (DbException\*)  
*Database::Database* *Database* is a singleton.
- QVariant `value` (const QSqlQuery &q, const QString &champ)  
*Database::valeur* Value of database field.

### Protected Attributes

- QSettings \* `_settings`  
*settings*
- QSqlDatabase `mDatabase`  
*contains Database*
- QList< `Database * >` `_instances`  
*List of instances.*

### Static Protected Attributes

- static `Database * _instance` = 0  
*Instance.*
- static bool `_dbInstance` = 0  
*an instance of db is open*
- static bool `isOpen` = false  
*Database is open.*



### 5.31.1 Detailed Description

The [Database](#) class Master class for all database access.

#### Author

Antoine de Roquemaurel

### 5.31.2 Member Function Documentation

#### 5.31.2.1 void Database::executeFile ( QString *pName* )

Database::executerFichier Exeute a specified file named *pName*

##### Parameters

<i>pNom</i>	File name
-------------	-----------

#### 5.31.2.2 Database \* Database::instance ( ) throw DbException \*) [static]

Database::getInstance Return an instance of [Database](#).

##### Returns

Instance of [Database](#)

#### 5.31.2.3 QString Database::lastError ( const QSqlQuery & *q* ) [inline]

[Database::lastError](#) Return an error message on the last error occurred during the SQL request *q*

##### Parameters

<i>q</i>	SQL request
----------	-------------

##### Returns

an error message

#### 5.31.2.4 void Database::setDatabase ( QSqlDatabase *sql* )

[Database::setDatabase](#) Set database.

##### Parameters

<i>sql</i>	The new database
------------	------------------

#### 5.31.2.5 QVariant Database::value ( const QSqlQuery & *q*, const QString & *champ* ) [protected]

Database::valeur Value of database field.

##### Parameters

<i>q</i>	Query
<i>champ</i>	Field

#### Returns

The value

The documentation for this class was generated from the following files:

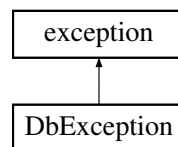
- src/database/database.h
- src/database/database.cpp

## 5.32 DbException Class Reference

The [DbException](#) class for database exception : queries, db file, ...

```
#include <dbexception.h>
```

Inheritance diagram for DbException:



### Public Member Functions

- [DbException](#) (const QString fct, const QString fctName, const QString logError, float errorCode)  
[DbException::DbException](#). Construct a [DbException](#).
- virtual [~DbException](#) () throw ()  
[~DbException](#)
- void [popupMessage](#) (QWidget \*parent)  
[DbException::popupMessage](#). Display a popup message with the message error.

#### 5.32.1 Detailed Description

The [DbException](#) class for database exception : queries, db file, ...

#### Author

Antoine de Roquemaurel

#### 5.32.2 Constructor & Destructor Documentation

5.32.2.1 [DbException::DbException](#) ( const QString *fct*, const QString *fctName*, const QString *logError*, float *errorCode* )

[DbException::DbException](#). Construct a [DbException](#).

## Parameters

<i>userError</i>	ClassName of error
<i>fctName</i>	Function name
<i>logError</i>	Message error
<i>errorCode</i>	Code of error

## 5.32.3 Member Function Documentation

5.32.3.1 void DbException::popupMessage ( QWidget \* *parent* )

[DbException::popupMessage](#). Display a popup message with the message error.

## Parameters

<i>parent</i>	
---------------	--

The documentation for this class was generated from the following files:

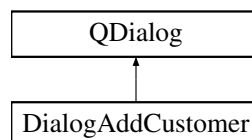
- src/exceptions/dbexception.h
- src/exceptions/dbexception.cpp

## 5.33 DialogAddCustomer Class Reference

The [DialogAddCustomer](#) class Window to add or modify a [Customer](#).

```
#include <dialogaddcustomer.h>
```

Inheritance diagram for DialogAddCustomer:



## Public Slots

- void [checkFields](#) ()  
*setFieldsComponents*

## Public Member Functions

- [DialogAddCustomer](#) (int id=0, QWidget \*parent=0)  
*DialogAddCustomer Construct a window to add/modify a Customer.*
- void [fillFields](#) ()  
*fillFields If the Customer exists, fill line edits with the data of the current Customer*
- void [accept](#) ()  
*accept Valid data inputed by user and add these data in Database*
- void [reject](#) ()  
*reject Cancel the operation and close the windows*

### 5.33.1 Detailed Description

The [DialogAddCustomer](#) class Window to add or modify a [Customer](#).

#### Author

Cédric Rohaut

### 5.33.2 Constructor & Destructor Documentation

#### 5.33.2.1 [DialogAddCustomer::DialogAddCustomer](#) ( int *id* = 0, QWidget \* *parent* = 0 ) [explicit]

[DialogAddCustomer](#) Construct a window to add/modify a [Customer](#).

#### Parameters

<i>id</i>	<a href="#">Customer</a> id
<i>parent</i>	QWidget parent

The documentation for this class was generated from the following files:

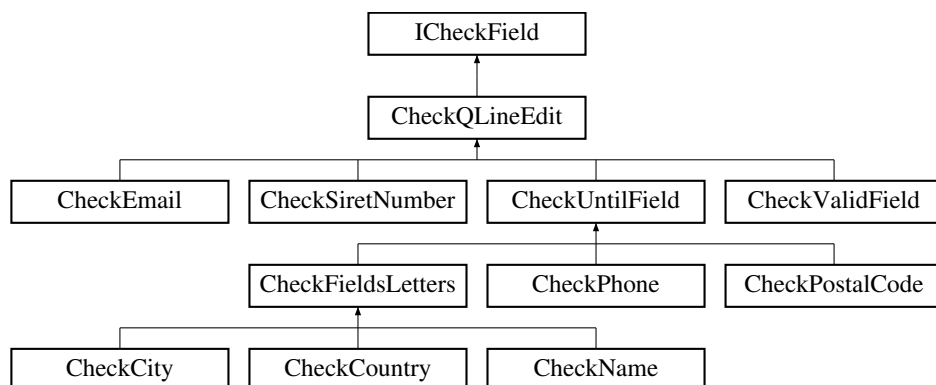
- src/dialogs/dialogaddcustomer.h
- src/dialogs/dialogaddcustomer.cpp

## 5.34 ICheckField Class Reference

The [ICheckField](#) class Interface to check fields validity.

```
#include <icheckfield.h>
```

Inheritance diagram for ICheckField:



### Public Member Functions

- virtual bool [check](#) (QString text)=0  
*check* Check if the field (line edit) is valid Return TRUE if the field is valid, else FALSE

### 5.34.1 Detailed Description

The [ICheckField](#) class Interface to check fields validity.

### 5.34.2 Member Function Documentation

#### 5.34.2.1 virtual bool ICheckField::check ( QString text ) [pure virtual]

check Check if the field (line edit) is valid Return TRUE if the field is valid, else FALSE

#### Returns

boolean

Implemented in [CheckEmail](#), [CheckPhone](#), [CheckPostalCode](#), [CheckSiretNumber](#), [CheckUntilField](#), [CheckFieldsLetters](#), and [CheckValidField](#).

The documentation for this class was generated from the following file:

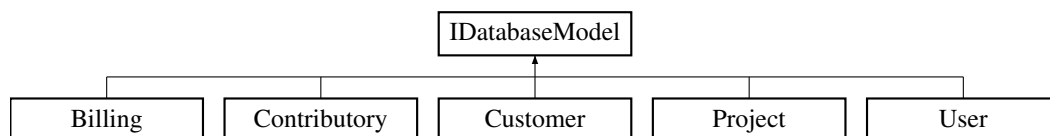
- src/widgets/checkfields/ichckfield.h

## 5.35 IDatabaseModel Class Reference

The [IDatabaseModel](#) class.

```
#include <idatabasemodel.h>
```

Inheritance diagram for IDatabaseModel:



### Public Member Functions

- virtual [~IDatabaseModel](#) ()  
*~IDatabaseModel Remove an instance of [IDatabaseModel](#)*
- virtual void [commit](#) ()=0  
*[IDatabaseModel::commit](#) Update or insert data into the database.*
- virtual void [hydrat](#) (int id)=0  
*[IDatabaseModel::hydrat](#) Get data of the element which is specified by the identify id from the database.*
- virtual void [remove](#) ()=0  
*[IDatabaseModel::remove](#) Remove the current element in the database.*
- int [getId](#) () const  
*[IDatabaseModel::getId](#) Return the identify of the element of the database.*
- void [setId](#) (int id)  
*[IDatabaseModel::setId](#) Replace the current identify by id*
- bool [isToRemoved](#) () const  
*toRemoved return if object must be removed.*
- void [setToRemoved](#) (bool toRemoved)  
*setToRemoved Change the flag for removed object*

## Protected Attributes

- `int _id`  
*Element identify.*
- `bool _toRemoved`  
*Flag to know if the object must be removed.*

### 5.35.1 Detailed Description

The `IDatabaseModel` class.

Author

Antoine de Roquemaurel

### 5.35.2 Member Function Documentation

5.35.2.1 `int IDatabaseModel::getId ( ) const` `[inline]`

`IDatabaseModel::getId` Return the identify of the element of the database.

Returns

identity

5.35.2.2 `virtual void IDatabaseModel::hydrat ( int id )` `[pure virtual]`

`IDatabaseModel::hydrat` Get data of the element which is specified by the identify *id* from the database.

Parameters

<i>id</i>	
-----------	--

Implemented in `Project`, `Billing`, `Contributory`, `User`, and `Customer`.

5.35.2.3 `bool IDatabaseModel::isToRemoved ( ) const` `[inline]`

`toRemoved` return if object must be removed.

Returns

5.35.2.4 `void IDatabaseModel::setId ( int id )` `[inline]`

`IDatabaseModel::setId` Replace the current identify by *id*

Parameters

<i>id</i>	New identify
-----------	--------------

5.35.2.5 `void IDatabaseModel::setToRemoved ( bool toRemoved )` `[inline]`

`setToRemoved` Change the flag for removed object

## Parameters

<i>toRemoved</i>	The new flag
------------------	--------------

The documentation for this class was generated from the following file:

- src/models/databasemodel.h

## 5.36 ItemType Class Reference

The [ItemType](#) class Item type model.

```
#include <itemtype.h>
```

### Public Member Functions

- [ItemType](#) (int type, QString name)  
*ItemType::ItemType* Construct an Item type.
- QString [getName](#) () const  
*ItemType::getName* Get item name.
- [IDatabaseModel](#) \* [getModel](#) (int id)  
*ItemType::getModel* Get the databasemodel of the [ItemType](#) according to this identity id
- void [setName](#) (const QString &name)  
*ItemType::setName* Modify the item name.
- int [getType](#) () const  
*ItemType::getType* Get the type of the current item.
- void [setType](#) (int type)  
*ItemType::setType* Modify the type of the current item.

### Static Public Attributes

- static const int [CUSTOMER](#) = 0  
*constant value assigned to Customer*
- static const int [PROJECT](#) = 1  
*constant value assigned to Project*
- static const int [BILLING](#) = 2  
*constant value assigned to Billing*
- static const int [QUOTE](#) = 3  
*constant value assigned to Quote*

#### 5.36.1 Detailed Description

The [ItemType](#) class Item type model.

#### 5.36.2 Constructor & Destructor Documentation

##### 5.36.2.1 ItemType::ItemType ( int type, QString name )

[ItemType::ItemType](#) Construct an Item type.

## Parameters

<i>type</i>	Type of the item
<i>name</i>	Name of the item

### 5.36.3 Member Function Documentation

#### 5.36.3.1 `IDatabaseModel * ItemType::getModel ( int id )`

[ItemType::getModel](#) Get the databasemodel of the [ItemType](#) according to this identity *id*

## Parameters

<i>id</i>	Item type identity
-----------	--------------------

## Returns

database model

#### 5.36.3.2 `QString ItemType::getName ( ) const`

[ItemType::getName](#) Get item name.

## Returns

item name

#### 5.36.3.3 `int ItemType::getType ( ) const`

[ItemType::getType](#) Get the type of the current item.

## Returns

type of the current item

#### 5.36.3.4 `void ItemType::setName ( const QString & name )`

[ItemType::setName](#) Modify the item name.

## Parameters

<i>name</i>	New Item name
-------------	---------------

#### 5.36.3.5 `void ItemType::setType ( int type )`

[ItemType::setType](#) Modify the type of the current item.

## Parameters

<i>type</i>	New item type
-------------	---------------

The documentation for this class was generated from the following files:

- `src/itemtype.h`
- `src/itemtype.cpp`



## 5.37 Log Class Reference

The [Log](#) class for Simple management of log.

```
#include <log.h>
```

### Public Member Functions

- [~Log](#) ()  
*[Log::~~Log](#).*
- void [write](#) (const QString text)  
*[Log::write](#). Write log message in file.*
- [Log](#) ()  
*[Log::Log](#). [Log](#) is a singleton.*

### Static Public Member Functions

- static [Log](#) & [instance](#) (TypeLog type=INFO)  
*[Log::instance](#). Return the instance of logger.*

### Friends

- [Log](#) & [operator<<](#) ([Log](#) &logger, const QString &text)  
*[operator<<](#) for log writing*

#### 5.37.1 Detailed Description

The [Log](#) class for Simple management of log.

#### 5.37.2 Member Function Documentation

##### 5.37.2.1 [Log](#) & [Log::instance](#) ( TypeLog *type* = INFO ) [static]

[Log::instance](#). Return the instance of logger.

##### Parameters

<i>type</i>	Type of log : WARNING, INFO, ERROR
-------------	------------------------------------

##### Returns

Instance of logger.

##### 5.37.2.2 void [Log::write](#) ( const QString *text* )

[Log::write](#). Write log message in file.

##### Parameters

<i>text</i>	
-------------	--

### 5.37.3 Friends And Related Function Documentation

#### 5.37.3.1 Log& operator<< ( Log & *logger*, const QString & *text* ) [friend]

operator << for log writing

##### Parameters

<i>logger</i>	Instance of Logger
<i>text</i>	Text to write

##### Returns

New logger.

The documentation for this class was generated from the following files:

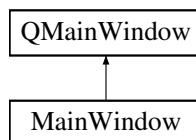
- src/log.h
- src/log.cpp

## 5.38 MainWindow Class Reference

The [MainWindow](#) class Main Window of the software.

```
#include <mainwindow.h>
```

Inheritance diagram for MainWindow:



### Public Slots

- void [addCustomer](#) ()  
*MainWindow::addCustomer* open window to add a new customer.
- void [editCustomer](#) ()  
*MainWindow::editCustomer* open window to modify a customer.
- void [removeCustomer](#) ()  
*MainWindow::removeCustomer* open a popup to confirm the deletion of a customer, if ok remove the customer.
- void [addQuote](#) ()  
*MainWindow::addQuote* open window to add a new quote.
- void [editUser](#) ()  
*MainWindow::editUser* modify the user.
- void [search](#) (QString s)  
*MainWindow::search* launch a new search.
- void [search](#) ()  
*MainWindow::search* call [search\(QString\)](#)
- void [newProject](#) ()

- [\*MainWindow::newProject\*](#) *Create a new project for a customer.*
- void [removeProject](#) (void)  
[\*MainWindow::removeProject\*](#) *Remove a project for a customer.*
- void [editProject](#) (void)  
[\*MainWindow::editProject\*](#) *Modify the customer project.*
- void [aboutQt](#) ()  
[\*MainWindow::aboutQt\*](#) *show Qt's details.*
- void [aboutFact](#) ()  
[\*MainWindow::aboutFact\*](#) *show FACT's details (FACT team)*
- void [aboutFactDev](#) ()  
[\*MainWindow::aboutFactDev\(\)\*](#) *show FactDev's details (FactDev Software)*
- void [aboutIcons](#) ()  
[\*MainWindow::aboutIcons\(\)\*](#) *show icons's details.*

## Public Member Functions

- [MainWindow](#) (QWidget \*parent=0)  
[\*MainWindow\*](#): *Construct a window.*
- int [getCurrentCustomerId](#) ()  
[\*MainWindow::getCurrentCustomerId\*](#) *get the selected customer.*
- int [getCurrentProjectId](#) ()  
[\*MainWindow::getCurrentProjectId\*](#) *get the selected project id.*
- QString [getCurrentCustomerName](#) ()  
[\*MainWindow::getCurrentCustomerName\*](#) *get the selected customer name in the customers' table.*
- QString [getCurrentProjectName](#) ()  
[\*MainWindow::getCurrentProjectName\*](#) *get the selected project name in the table of projects.*
- bool [isTreeRoot](#) ()  
[\*MainWindow::isTreeRoot\*](#) *return if the node selected in the tree is the root.*
- bool [isCustomerItemTree](#) ()  
[\*MainWindow::isCustomerItemTree\*](#) *return if the node selected in the tree is a customer.*
- bool [isProjectItemTree](#) ()  
[\*MainWindow::isProjectItemTree\*](#) *return if the node selected in the tree is a project.*
- bool [isQuoteItemTree](#) ()
- void [demo](#) ()

### 5.38.1 Detailed Description

The [MainWindow](#) class Main Window of the software.

Author

Everybody

### 5.38.2 Constructor & Destructor Documentation

#### 5.38.2.1 [MainWindow::MainWindow](#) ( [QWidget](#) \* *parent* = 0 ) [explicit]

[MainWindow](#): Construct a window.

## Parameters

<i>parent</i>	
---------------	--

### 5.38.3 Member Function Documentation

5.38.3.1 void MainWindow::addQuote ( ) [slot]

[MainWindow::addQuote](#) open window to add a new quote.

See also

[AddQuoteDialog](#)

5.38.3.2 void MainWindow::editUser ( ) [slot]

[MainWindow::editUser](#) modify the user.

See also

[UserDataDialog](#)

5.38.3.3 int MainWindow::getCurrentCustomerId ( )

[MainWindow::getCurrentCustomerId](#) get the selected customer.

Returns

id of the selected customer

5.38.3.4 QString MainWindow::getCurrentCustomerName ( )

[MainWindow::getCurrentCustomerName](#) get the selected customer name in the customers' table.

Returns

name of the selected customer

5.38.3.5 int MainWindow::getCurrentProjectId ( )

[MainWindow::getCurrentProjectId](#) get the selected project id.

Returns

id of the selected project

5.38.3.6 QString MainWindow::getCurrentProjectName ( )

[MainWindow::getCurrentProjectName](#) get the selected project name in the table of projects.

Returns

name of the selected project

#### 5.38.3.7 bool MainWindow::isCustomerItemTree ( )

[MainWindow::isCustomerItemTree](#) return if the node selected in the tree is a customer.

##### Returns

boolean

#### 5.38.3.8 bool MainWindow::isProjectItemTree ( )

[MainWindow::isProjectItemTree](#) return if the node selected in the tree is a project.

##### Returns

boolean

#### 5.38.3.9 bool MainWindow::isTreeRoot ( )

[MainWindow::isTreeRoot](#) return if the node selected in the tree is the root.

##### Returns

boolean

#### 5.38.3.10 void MainWindow::newProject ( ) [slot]

[MainWindow::newProject](#) Create a new project for a customer.

##### See also

[AddProjectDialog](#)

#### 5.38.3.11 void MainWindow::search ( QString s ) [slot]

[MainWindow::search](#) launch a new search.

##### Parameters

s	text in field
---	---------------

The documentation for this class was generated from the following files:

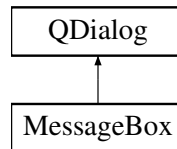
- src/mainwindow.h
- src/mainwindow.cpp

## 5.39 MessageBox Class Reference

The [MessageBox](#) class Information window with message.

```
#include <messagebox.h>
```

Inheritance diagram for MessageBox:



## Public Member Functions

- [MessageBox](#) (QWidget \*parent=0)  
[MessageBox::MessageBox](#) Construt a **MessageBox**
- void [aboutFact](#) ()  
[MessageBox::aboutFact](#) Defines FACT team information.
- void [aboutFactDev](#) ()  
[MessageBox::aboutFactDev](#) Defines FactDev software information.
- void [aboutIcons](#) ()  
[MessageBox::aboutIcons](#) Defines icons theme information.
- void [setImage](#) (QString img, int width=128, int height=128)  
[MessageBox::setImage](#) Add the icon img to the current window.
- void [setText](#) (QString txt)  
[MessageBox::setText](#) Add the text txt to the current window.

## Static Public Member Functions

- static void [showAboutFact](#) ()  
[MessageBox::showAboutFact](#) Shows window about FACT team.
- static void [showAboutFactDev](#) ()  
[MessageBox::showAboutFactDev](#) Shows window about FactDev software.
- static void [showAboutIcons](#) ()  
[MessageBox::showAboutIcons](#) Shows about icons theme of FactDev software.

### 5.39.1 Detailed Description

The [MessageBox](#) class Information window with message.

Author

Florent Berbie

### 5.39.2 Constructor & Destructor Documentation

#### 5.39.2.1 [MessageBox::MessageBox](#) ( QWidget \* parent = 0 ) [explicit]

[MessageBox::MessageBox](#) Construt a **MessageBox**

Parameters

<i>parent</i>	
---------------	--

### 5.39.3 Member Function Documentation

#### 5.39.3.1 void [MessageBox::setImage](#) ( QString *img*, int *width* = 128, int *height* = 128 )

[MessageBox::setImage](#) Add the icon *img* to the current window.

## Parameters

<i>img</i>	Icon
<i>width</i>	Icon width (default: 128)
<i>height</i>	Icon height (default: 128)

5.39.3.2 void `MessageBox::setText ( QString txt )`

[MessageBox::setText](#) Add the text *txt* to the current window.

## Parameters

<i>txt</i>	Text inside the current window
------------	--------------------------------

The documentation for this class was generated from the following files:

- `src/dialogs/messagebox/messagebox.h`
- `src/dialogs/messagebox/messagebox.cpp`

## 5.40 Parameters Class Reference

### Public Member Functions

- [Parameters](#) ()  
*Parameters* Construct a new parameters.

### Static Public Attributes

- static const QString [DB\\_FILENAME](#) = "database.db"  
*DB\_FILENAME* The database sqlite file name.
- static const double [VERSION](#) = 0.2  
*VERSION* Version number of software.

#### 5.40.1 Member Data Documentation

5.40.1.1 const QString `Parameters::DB_FILENAME = "database.db"` [static]

`DB_FILENAME` The database sqlite file name.

[Database](#) file name

5.40.1.2 const double `Parameters::VERSION = 0.2` [static]

`VERSION` Version number of software.

Application version

The documentation for this class was generated from the following files:

- `src/parameters.h`
- `src/parameters.cpp`

## 5.41 Popup Class Reference

Class for display popup quickly.

```
#include <popup.h>
```

### Static Public Member Functions

- static void [toImplement](#) (QString, QWidget \*)  
*[Popup::toImplement](#) Method to display a critical message : feature is not implemented now.*

### 5.41.1 Detailed Description

Class for display popup quickly.

Author

Antoine de Roquemaurel

The documentation for this class was generated from the following files:

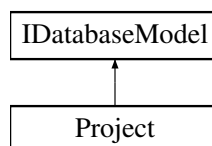
- src/widgets/popup.h
- src/widgets/popup.cpp

## 5.42 Project Class Reference

The [Project](#) class : [Project](#) linked to a [Customer](#).

```
#include <project.h>
```

Inheritance diagram for Project:



### Public Member Functions

- [Project](#) ()  
*[Project::Project](#) Construct a [Project](#).*
- [Project](#) (QString name)  
*[Project::project](#) Construct a project with a name.*
- [Project](#) (int id)  
*[Project::Project](#) Construct a [Project](#) which is specified by an id*
- virtual [~Project](#) ()  
*[~Project](#) Destruct project object*
- void [commit](#) ()  
*[Project::commit](#) Update project data in the database.*
- void [hydrat](#) (int id)  
*[Project::hydrat](#) Insert project data which is specified by id in the database.*



- void [remove](#) ()  
*[Project::remove](#) Remove the current project.*
- QString [getName](#) () const  
*[Project::getName](#) Return the project name.*
- void [setName](#) (const QString &name)  
*[Project::setName](#) Modify the project name*
- QString [getDescription](#) () const  
*[Project::getDescription](#) Return a project description.*
- void [setDescription](#) (const QString &description)  
*[Project::setDescription](#) Modify the project description*
- QDate [getBeginDate](#) () const  
*[Project::getBeginDate](#) return the date of creation of the [Project](#)*
- void [setBeginDate](#) (QDate beginDate)  
*[Project::setBeginDate](#) Modify beginDate of a [Project](#)*
- QDate [getEndDate](#) ()  
*[Project::getEndDate](#) Return the endDate of the [Project](#)*
- void [setEndDate](#) (QDate endDate)  
*[Project::setEndDate](#) Modify endDate of [Project](#)*
- double [getDailyRate](#) () const  
*[Project::getDailyRate](#) Return the daily rate estimated for this project.*
- void [setDailyRate](#) (double dailyRate)  
*[Project::setDailyRate](#) Modify the daily rate dailyRate of the current project.*
- QSharedPointer< [Customer](#) > [getCustomer](#) () const  
*[Project::getCustomer](#) Return the reference to the customer linked to this project.*
- void [setCustomer](#) (QSharedPointer< [Customer](#) > customer)  
*[Project::setCustomer](#) Modify the customer linked to this project.*

### Static Public Attributes

- static const double [LEGAL\\_RATE](#) = 9.61  
*The Legal hourly Rate.*
- static const int [NB\\_DAILY\\_HOURS](#) = 7  
*The number of daily hours in a day.*

### Additional Inherited Members

#### 5.42.1 Detailed Description

The [Project](#) class : [Project](#) linked to a [Customer](#).

#### Author

Florent Berbie

#### See also

[IDatabaseModel](#)

#### 5.42.2 Constructor & Destructor Documentation

##### 5.42.2.1 [Project::Project](#) ( int *id* )

[Project::Project](#) Construct a [Project](#) which is specified by an *id*

## Parameters

<i>id</i>	
-----------	--

### 5.42.3 Member Function Documentation

#### 5.42.3.1 QDate Project::getBeginDate ( ) const

[Project::getBeginDate](#) return the date of creation of the [Project](#)

## Returns

the begin date of the [Project](#)

#### 5.42.3.2 QSharedPointer< Customer > Project::getCustomer ( ) const

[Project::getCustomer](#) Return the reference to the customer linked to this project.

## Returns

customer linked to this project

#### 5.42.3.3 double Project::getDailyRate ( ) const

[Project::getDailyRate](#) Return the daily rate estimated for this project.

## Returns

the daily rate linket to the current project

#### 5.42.3.4 QString Project::getDescription ( ) const

[Project::getDescription](#) Return a project description.

## Returns

project description

#### 5.42.3.5 QDate Project::getEndDate ( )

[Project::getEndDate](#) Return the *endDate* of the [Project](#)

## Returns

the end date of the project

#### 5.42.3.6 QString Project::getName ( ) const

[Project::getName](#) Return the project name.

## Returns

project name

5.42.3.7 void Project::hydrat ( int *id* ) [virtual]

[Project::hydrat](#) Insert project data which is specified by *id* in the database.

## Parameters

<i>id</i>	<a href="#">Project</a> identify
-----------	----------------------------------

Implements [IDatabaseModel](#).

5.42.3.8 void [Project::setBeginDate](#) ( [QDate](#) *beginDate* )

[Project::setBeginDate](#) Modify *beginDate* of a [Project](#)

## Parameters

<i>beginDate</i>	the new date of creation of the project
------------------	---

5.42.3.9 void [Project::setCustomer](#) ( [QSharedPointer](#)< [Customer](#) > *customer* )

[Project::setCustomer](#) Modify the *customer* linked to this project.

## Parameters

<i>customer</i>	New customer associated to this project
-----------------	---

5.42.3.10 void [Project::setDailyRate](#) ( [double](#) *dailyRate* )

[Project::setDailyRate](#) Modify the daily rate *dailyRate* of the current project.

## Parameters

<i>dailyRate</i>	New daily rate associated to the current project
------------------	--

5.42.3.11 void [Project::setDescription](#) ( [const QString](#) & *description* )

[Project::setDescription](#) Modify the project *description*

## Parameters

<i>description</i>	New project description
--------------------	-------------------------

5.42.3.12 void [Project::setEndDate](#) ( [QDate](#) *endDate* )

[Project::setEndDate](#) Modify *endDate* of [Project](#)

## Parameters

<i>endDate</i>	the new end date of the project
----------------	---------------------------------

5.42.3.13 void [Project::setName](#) ( [const QString](#) & *name* )

[Project::setName](#) Modify the project *name*

## Parameters

---

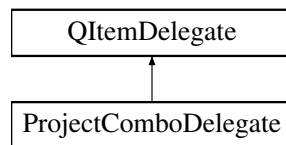
<i>name</i>	<a href="#">Project</a> name
-------------	------------------------------

The documentation for this class was generated from the following files:

- src/models/project.h
- src/models/project.cpp

## 5.43 ProjectComboDelegate Class Reference

Inheritance diagram for ProjectComboDelegate:



### Public Member Functions

- **ProjectComboDelegate** (QSharedPointer< [Customer](#) > c, QObject \*parent=0)
- QWidget \* **createEditor** (QWidget \*parent, const QStyleOptionViewItem &option, const QModelIndex &index) const
- void **setEditorData** (QWidget \*editor, const QModelIndex &index) const
- void **setModelData** (QWidget \*editor, QAbstractItemModel \*model, const QModelIndex &index) const
- void **updateEditorGeometry** (QWidget \*editor, const QStyleOptionViewItem &option, const QModelIndex &index) const
- void **paint** (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const

The documentation for this class was generated from the following files:

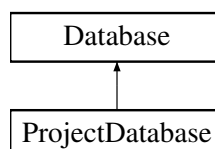
- src/widgets/delegates/projectcombodelegate.h
- src/widgets/delegates/projectcombodelegate.cpp

## 5.44 ProjectDatabase Class Reference

The [ProjectDatabase](#) class [Project](#) table database.

```
#include <projectdatabase.h>
```

Inheritance diagram for ProjectDatabase:



### Public Member Functions

- [Project](#) \* [getProject](#) (const int pld)  
*ProjectDatabase::getProject* Get informations about the project identified by 'pld'.

- int [addProject](#) (const [Project](#) &)  
*[ProjectDatabase::addProject](#) Add the project 'pProject' to the database.*
- void [updateProject](#) (const [Project](#) &)  
*[ProjectDatabase::updateProject](#) Update informations about the project.*
- void [removeProject](#) (const int pld)  
*[removeProject](#) Remove the project with the id 'pld'*
- int [getNbProjects](#) ()  
*[ProjectDatabase::getNbProjects](#) Return the number of projects existing.*
- int [getNbProjectsForACustomer](#) (const int pld)  
*[ProjectDatabase::getNbProjectsForACustomer](#) Return the number of projects existing for an identify customer pld*
- QMap< int, [Project](#) > [getProjectsOfCustomer](#) (QSharedPointer< [Customer](#) > c)  
*[getProjectsOfCustomer](#) Return all projects of a customer*
- QStandardItemModel \* [getProjectsTable](#) (const int pld) throw (DbException\*)  
*[getProjectsTable](#) Return all projects of a customer in QStandardItemModel*
- [Project](#) \* [getProject](#) (QStringQuery &q)  
*[getProject](#) Obtain a project without new query*

## Static Public Member Functions

- static [ProjectDatabase](#) \* [instance](#) () throw (DbException\*)  
*[ProjectDatabase::getInstance](#) Return an instance of [ProjectDatabase](#).*

## Additional Inherited Members

### 5.44.1 Detailed Description

The [ProjectDatabase](#) class [Project](#) table database.

#### Author

Florent Berbie

#### See also

[Database](#)  
[Project](#)

### 5.44.2 Member Function Documentation

#### 5.44.2.1 int ProjectDatabase::addProject ( const Project & pProject )

[ProjectDatabase::addProject](#) Add the project 'pProject' to the database.

#### Returns

project id

#### 5.44.2.2 int ProjectDatabase::getNbProjects ( )

[ProjectDatabase::getNbProjects](#) Return the number of projects existing.

#### Returns

number of projects

5.44.2.3 `int ProjectDatabase::getNbProjectsForACustomer ( const int pld )`

[ProjectDatabase](#):`getNbProjectsForACustomer` Return the number of projects existing for an identify customer *pld*

## Parameters

<i>pld</i>	Project id
------------	------------

## Returns

number of projects

5.44.2.4 Project \* ProjectDatabase::getProject ( const int *pld* )

ProjectDatabase::getProject Get informations about the project identified by 'pld'.

## Parameters

<i>pld</i>	project
------------	---------

## Returns

the project

5.44.2.5 Project \* ProjectDatabase::getProject ( QSqlQuery & *q* )

getProject Obtain a project without new query

## Parameters

<i>q</i>	The query to use
----------	------------------

## Returns

The project linked to q

5.44.2.6 QMap< int, Project > ProjectDatabase::getProjectsOfCustomer ( QSharedPointer< Customer > *c* )

getProjectsOfCustomer Return all projects of a customer

## Parameters

<i>c</i>	The customer
----------	--------------

## Returns

All projects of c with id in key

5.44.2.7 QStandardItemModel \* ProjectDatabase::getProjectsTable ( const int *pld* ) throw DbException \*)

getProjectsTable Return all projects of a customer in QStandardItemModel

## Parameters

<i>filter</i>	Select only projects who are specified by <i>filter</i>
---------------	---

## Returns

QStandardItemModel an item model for QTableView



#### 5.44.2.8 ProjectDatabase \* ProjectDatabase::instance ( ) throw DbException \*) [static]

ProjectDatabase::getInstance Return an instance of [ProjectDatabase](#).

##### Returns

Instance of [ProjectDatabase](#)

#### 5.44.2.9 void ProjectDatabase::removeProject ( const int pld )

removeProject Remove the project with the id 'pld'

##### Parameters

<i>pld</i>	project id
------------	------------

The documentation for this class was generated from the following files:

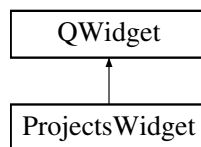
- src/database/projectdatabase.h
- src/database/projectdatabase.cpp

## 5.45 ProjectsWidget Class Reference

The [ProjectsWidget](#) class Actions on [Project](#).

```
#include <projectswidget.h>
```

Inheritance diagram for ProjectsWidget:



### Public Slots

- void [newProject](#) ()  
*[ProjectsWidget::newProject](#) Event which sends a signal to add a new project.*
- void [editSelectedProject](#) ()  
*[ProjectsWidget::editSelectedProject](#) Event which sends a signal to edit the project selected.*
- void [removeSelectedProject](#) ()  
*[ProjectsWidget::removeSelectedProject](#) Event which sends a signal to remove the project selected.*

### Signals

- void [addProject](#) ()  
*[ProjectsWidget::addProject](#) Add a new project to the current [Customer](#).*
- void [editProject](#) ()  
*[ProjectsWidget::editProject](#) Edit the current [Customer](#) selected.*
- void [removeProject](#) ()  
*[ProjectsWidget::removeProject](#) Remove the current [Customer](#) selected.*

## Public Member Functions

- [ProjectsWidget](#) (QWidget \*parent=0)  
*ProjectsWidget::ProjectsWidget* Construct a *ProjectsWidget*.

### 5.45.1 Detailed Description

The [ProjectsWidget](#) class Actions on [Project](#).

Author

Florent Berbie

### 5.45.2 Constructor & Destructor Documentation

#### 5.45.2.1 ProjectsWidget::ProjectsWidget ( QWidget \* parent = 0 ) [explicit]

[ProjectsWidget::ProjectsWidget](#) Construct a [ProjectsWidget](#).

Parameters

<i>parent</i>	
---------------	--

The documentation for this class was generated from the following files:

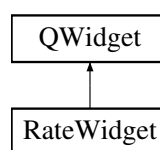
- src/widgets/projectswidget.h
- src/widgets/projectswidget.cpp

## 5.46 RateWidget Class Reference

Class for display Rate.

```
#include <ratewidget.h>
```

Inheritance diagram for RateWidget:



## Public Slots

- void [setDailyRate](#) ()  
*setDailyRate* Set a new value for the daily rate
- void [setHourlyRate](#) ()  
*setHourlyRate* Set a new value for the hourly rate

## Public Member Functions

- [RateWidget](#) (QWidget \*parent=0)  
*RateWidget::RateWidget* Construct a rate widget.
- void [initRate](#) ()

- [\*RateWidget::initRate\*](#) *Initialize the rate.*
- void [\*setWidgetDailyRateValue\*](#) (double value)
  - [\*RateWidget::setWidgetDailyRateValue\*](#) *Modify the value of the daily rate spin box component.*
- double [\*getDailyRate\*](#) ()
  - [\*RateWidget::getDailyRate\*](#) *Get the daily rate.*
- double [\*getHourlyRate\*](#) ()
  - [\*RateWidget::getHourlyRate\*](#) *Get the hourly rate.*
- void [\*setWidgetHourlyRateValue\*](#) (double value)
  - [\*RateWidget::setWidgetHourlyRateValue\*](#) *Modify the value of the hourly rate spin box component.*
- void [\*updateConversionRate\*](#) ()
  - [\*updateConversionRate\*](#) *Update daily rate or hourly rate*

### 5.46.1 Detailed Description

Class for display Rate.

Author

Florent Berbie

### 5.46.2 Constructor & Destructor Documentation

5.46.2.1 [\*RateWidget::RateWidget\*](#) ( *QWidget \* parent = 0* ) [*explicit*]

[\*RateWidget::RateWidget\*](#) Construct a rate widget.

Parameters

<i>parent</i>	The QWidget parent
---------------	--------------------

### 5.46.3 Member Function Documentation

5.46.3.1 double [\*RateWidget::getDailyRate\*](#) ( )

[\*RateWidget::getDailyRate\*](#) Get the daily rate.

Returns

The daily rate

5.46.3.2 double [\*RateWidget::getHourlyRate\*](#) ( )

[\*RateWidget::getHourlyRate\*](#) Get the hourly rate.

Returns

The hourly rate

5.46.3.3 void [\*RateWidget::setDailyRate\*](#) ( ) [*slot*]

[\*setDailyRate\*](#) Set a new value for the daily rate

## Parameters

<i>dailyRate</i>	The new daily rate
------------------	--------------------

## 5.46.3.4 void RateWidget::setHourlyRate ( ) [slot]

setHourlyRate Set a new value for the hourly rate

## Parameters

<i>hourlyRate</i>	The new hourly rate
-------------------	---------------------

## 5.46.3.5 void RateWidget::setWidgetDailyRateValue ( double value )

[RateWidget::setWidgetDailyRateValue](#) Modify the *value* of the daily rate spin box component.

## Parameters

<i>value</i>	New Value
--------------	-----------

## 5.46.3.6 void RateWidget::setWidgetHourlyRateValue ( double value )

[RateWidget::setWidgetHourlyRateValue](#) Modify the *value* of the hourly rate spin box component.

## Parameters

<i>value</i>	New value
--------------	-----------

The documentation for this class was generated from the following files:

- src/widgets/ratewidget.h
- src/widgets/ratewidget.cpp

## 5.47 Search Class Reference

The [Search](#) class.

```
#include <search.h>
```

### Public Member Functions

- [Search](#) ()  
[Search::Search](#) Construct a search.
- [~Search](#) ()  
[Search::Search](#) Destruct a search.
- QString [getFilter](#) ()  
[Search::getFilter](#) Return the search filter.
- bool [getSearchInCompanies](#) () const  
[Search::getSearchInCompanies](#) Return if we search a company.
- void [setSearchInCompanies](#) (bool [getSearchInCompanies](#))  
[Search::setSearchInCompanies](#) Modify the filter of search.
- bool [getSearchInReferentLastname](#) () const  
[Search::getSearchInReferentLastname](#) Return if we search a Last name referent.

- void [setSearchInReferentLastname](#) (bool [getSearchInReferentLastname](#))  
[Search::setSearchInReferentLastname](#) Modify the filter of search.
- bool [getGroupFilter](#) () const  
[Search::getGroupFilter](#) Return if the filter is activated.
- void [setGroupFilter](#) (bool [getGroupFilter](#))  
[Search::setGroupFilter](#) Modify if we active search filter.
- QString [getText](#) () const  
[Search::getText](#) Return sql portion of filter.
- void [setText](#) (const QString &[getText](#))  
[Search::setText](#) Modify sql portion.

### 5.47.1 Detailed Description

The [Search](#) class.

Author

Antoine de Roquemaurel

### 5.47.2 Member Function Documentation

#### 5.47.2.1 QString Search::getFilter ( )

[Search::getFilter](#) Return the search filter.

Returns

filter selected (sql portion)

#### 5.47.2.2 bool Search::getGroupFilter ( ) const

[Search::getGroupFilter](#) Return if the filter is activated.

Returns

boolean if search filter is activated

#### 5.47.2.3 bool Search::getSearchInCompanies ( ) const

[Search::getSearchInCompanies](#) Return if we search a company.

Returns

boolean if we search a company

#### 5.47.2.4 bool Search::getSearchInReferentLastname ( ) const

[Search::getSearchInReferentLastname](#) Return if we search a Last name referent.

Returns

boolean if search concerns the last name of referent

#### 5.47.2.5 QString Search::getText ( ) const

[Search::getText](#) Return sql portion of filter.

##### Returns

QString the sql portion

#### 5.47.2.6 void Search::setGroupFilter ( bool *getGroupFilter* )

[Search::setGroupFilter](#) Modify if we active search filter.

##### Parameters

<i>getGroupFilter</i>	Get if filter is activated
-----------------------	----------------------------

#### 5.47.2.7 void Search::setSearchInCompanies ( bool *getSearchInCompanies* )

[Search::setSearchInCompanies](#) Modify the filter of search.

##### Parameters

<i>getSearchInCompanies</i>	<a href="#">Search</a> in companies is concerned
-----------------------------	--

#### 5.47.2.8 void Search::setSearchInReferentLastname ( bool *getSearchInReferentLastname* )

[Search::setSearchInReferentLastname](#) Modify the filter of search.

##### Parameters

<i>getSearchInReferentLastname</i>	<a href="#">Search</a> in referent last name is concerned
------------------------------------	---

#### 5.47.2.9 void Search::setText ( const QString & *getText* )

[Search::setText](#) Modify sql portion.

##### Parameters

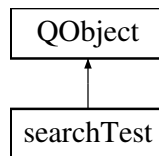
<i>getText</i>	Get sql portion
----------------	-----------------

The documentation for this class was generated from the following files:

- src/models/search.h
- src/models/search.cpp

## 5.48 searchTest Class Reference

Inheritance diagram for searchTest:



The documentation for this class was generated from the following files:

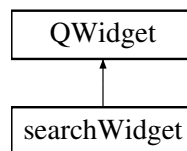
- tests/models/searchtest.h
- tests/models/searchtest.cpp

## 5.49 searchWidget Class Reference

Class for search in database.

```
#include <searchwidget.h>
```

Inheritance diagram for searchWidget:



### Public Slots

- void [search](#) (QString toSearch)  
*searchWidget::launch a search*
- void [getCustomerData](#) ()  
*searchWidget::getCustomerData Return data on the customer selected in the QTableView and display this data (First-name, Lastname, Company)*

### Signals

- void **selectCustomer** ()

### Public Member Functions

- [searchWidget](#) (QWidget \*parent=0)  
*searchWidget::searchWidget Construct a search widget*
- int [getCurrentCustomerId](#) ()  
*searchWidget::getCurrentCustomerId Return the id of the customer selected in the table*
- bool [isCustomerSelected](#) () const  
*isCustomerSelected Return TRUE if a customer is selected, else FALSE*
- void **selectCustomer** (int id)

#### 5.49.1 Detailed Description

Class for search in database.

## Author

Antoine de Roquemaurel

## 5.49.2 Constructor & Destructor Documentation

5.49.2.1 `searchWidget::searchWidget ( QWidget * parent = 0 )` `[explicit]`

[searchWidget::searchWidget](#) Construct a search widget

## Parameters

<i>parent</i>	The QWidget parent
---------------	--------------------

## 5.49.3 Member Function Documentation

5.49.3.1 `int searchWidget::getCurrentCustomerId ( )`

[searchWidget::getCurrentCustomerId](#) Return the id of the customer selected in the table

## Returns

id of the current customer

5.49.3.2 `bool searchWidget::isCustomerSelected ( ) const`

`isCustomerSelected` Return TRUE if a customer is selected, else FALSE

## Returns

boolean

5.49.3.3 `void searchWidget::search ( QString toSearch )` `[slot]`

`searchWidget::launch` a search

## Parameters

<i>toSearch</i>	The value to search
-----------------	---------------------

The documentation for this class was generated from the following files:

- `src/widgets/searchwidget.h`
- `src/widgets/searchwidget.cpp`

## 5.50 TestAdder< T > Class Template Reference

### Public Member Functions

- **TestAdder** (const QString &name)

The documentation for this class was generated from the following file:

- `tests/QTestRunner/testadder.cpp`



## 5.51 testadder Class Reference

The documentation for this class was generated from the following file:

- tests/QStringRunner/testadder.h

## 5.52 TestRunner Class Reference

### Public Member Functions

- template<typename T >  
char **RegisterTest** (QString name)
- int **RunAll** (int argc, char \*argv[])

### Static Public Member Functions

- static [TestRunner](#) & **Instance** ()

The documentation for this class was generated from the following files:

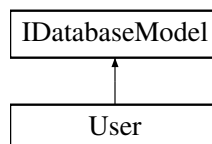
- tests/QStringRunner/testrunner.h
- tests/QStringRunner/testrunner.cpp

## 5.53 User Class Reference

The [User](#) class [User](#) of it application.

```
#include <user.h>
```

Inheritance diagram for User:



### Public Member Functions

- [User](#) ()  
*[User::User](#). Construct an [User](#).*
- [User](#) (int id)  
*[User::User](#). Construct a [User](#) with the identify id*
- void [commit](#) ()  
*[User::commit](#) Update user data in [User](#) table on the database.*
- void [hydrat](#) (int id=1)  
*[User::hydrat](#) Get data of the user who is specified by id from the database.*
- void [remove](#) ()  
*[remove](#) Remove the current [User](#)*
- QString [getFirstname](#) () const  
*[User::getFirstname](#) Return the user firstname.*

- void [setFirstname](#) (const QString &firstname)  
*User::setFirstname* Modify the user firstname
- QString [getLastname](#) () const  
*User::getLastname* Return the user lastname.
- void [setLastname](#) (const QString &lastname)  
*User::setLastname* Modify the user lastname
- QString [getCompany](#) () const  
*User::getCompany* Return the user company.
- void [setCompany](#) (const QString &company)  
*User::setCompany* Modify the user company name.
- QString [getTitle](#) () const  
*User::getTitle* Return a short description of [User](#) (company) activity.
- void [setTitle](#) (const QString &title)  
*User::setTitle* Modify the user/company activities description
- QString [getAddress](#) () const  
*User::getAddress* Return the company address (Number and name of street)
- void [setAddress](#) (const QString &address)  
*User::setAddress* Modify the user company address
- QString [getPostalCode](#) () const  
*User::getPostalCode* Return the postal code.
- void [setPostalCode](#) (const QString &postalCode)  
*User::setPostalCode* Modify the postal code postalCode
- QString [getCity](#) () const  
*User::getCity* Return the city.
- void [setCity](#) (const QString &city)  
*User::setCity* Modify the city
- QString [getEmail](#) () const  
*User::getEmail* Return the user professional email
- void [setEmail](#) (const QString &email)  
*User::setEmail* Modify the user professional email
- QString [getMobilePhone](#) () const  
*User::getMobilePhone* Return the number of the professional mobile phone.
- void [setMobilePhone](#) (const QString &mobilePhone)  
*User::setMobilePhone* Modify the number of the professional user mobile phone mobilePhone
- QString [getPhone](#) () const  
*User::getPhone* Return the number of the desktop phone.
- void [setPhone](#) (const QString &phone)  
*User::setPhone* Modify the number of the desktop phone
- QString [getNoSiret](#) () const  
*User::getNoSiret* Return the SIRET number (company registration number)
- void [setNoSiret](#) (const QString &noSiret)  
*User::setNoSiret* Modify the SIRET number (company registration number) noSiret

## Additional Inherited Members

### 5.53.1 Detailed Description

The [User](#) class [User](#) of it application.

Author

Florent Berbie

## 5.53.2 Constructor & Destructor Documentation

### 5.53.2.1 User::User ( int *id* )

[User::User](#). Construct a [User](#) with the identify *id*

Parameters

<i>id</i>	<a href="#">User</a> <i>id</i>
-----------	--------------------------------

## 5.53.3 Member Function Documentation

### 5.53.3.1 QString User::getAddress ( ) const

[User::getAddress](#) Return the company address (Number and name of street)

Returns

Address company

### 5.53.3.2 QString User::getCity ( ) const

[User::getCity](#) Return the city.

Returns

city

### 5.53.3.3 QString User::getCompany ( ) const

[User::getCompany](#) Return the user company.

Returns

New company name

### 5.53.3.4 QString User::getEmail ( ) const

[User::getEmail](#) Return the user professional *email*

Returns

professional email

### 5.53.3.5 QString User::getFirstname ( ) const

[User::getFirstname](#) Return the user firstname.

Returns

[User](#) firstname

#### 5.53.3.6 QString User::getLastname ( ) const

[User::getLastname](#) Return the user lastname.

##### Returns

[User](#) lastname

#### 5.53.3.7 QString User::getMobilePhone ( ) const

[User::getMobilePhone](#) Return the number of the professional mobile phone.

##### Returns

number of mobile phone

#### 5.53.3.8 QString User::getNoSiret ( ) const

[User::getNoSiret](#) Return the SIRET number (company registration number)

##### Returns

SIRET number

#### 5.53.3.9 QString User::getPhone ( ) const

[User::getPhone](#) Return the number of the desktop phone.

##### Returns

number of the desktop phone

#### 5.53.3.10 QString User::getPostalCode ( ) const

[User::getPostalCode](#) Return the postal code.

##### Returns

postal code

#### 5.53.3.11 QString User::getTitle ( ) const

[User::getTitle](#) Return a short description of [User](#) (company) activity.

##### Returns

a short description of user (company) activity

#### 5.53.3.12 void User::hydrat ( int *id* = 1 ) [virtual]

[User::hydrat](#) Get data of the user who is specified by *id* from the database.

## Parameters

<i>id</i>	User identify
-----------	---------------

Implements [IDatabaseModel](#).

5.53.3.13 void User::setAddress ( const QString & *address* )

[User::setAddress](#) Modify the user company *address*

## Parameters

<i>address</i>	Company address (name and number of street)
----------------	---

5.53.3.14 void User::setCity ( const QString & *city* )

[User::setCity](#) Modify the *city*

## Parameters

<i>city</i>	Company city address
-------------	----------------------

5.53.3.15 void User::setCompany ( const QString & *company* )

[User::setCompany](#) Modify the user *company* name.

## Parameters

<i>company</i>	New user company name
----------------	-----------------------

5.53.3.16 void User::setEmail ( const QString & *email* )

[User::setEmail](#) Modify the user professional *email*

## Parameters

<i>email</i>	The user professional email
--------------	-----------------------------

5.53.3.17 void User::setFirstname ( const QString & *firstname* )

[User::setFirstname](#) Modify the user *firstname*

## Parameters

<i>firstname</i>	New user firstname
------------------	--------------------

5.53.3.18 void User::setLastname ( const QString & *lastname* )

[User::setLastname](#) Modify the user *lastname*

## Parameters

<i>lastname</i>	New user lastname
-----------------	-------------------

5.53.3.19 void User::setMobilePhone ( const QString & *mobilePhone* )

[User::setMobilePhone](#) Modify the number of the professional user mobile phone *mobilePhone*

Parameters

<i>mobilePhone</i>	Number of the professional mobile phone
--------------------	---

5.53.3.20 void User::setNoSiret ( const QString & *noSiret* )

[User::setNoSiret](#) Modify the SIRET number (company registration number) *noSiret*

Parameters

<i>noSiret</i>	SIRET number
----------------	--------------

5.53.3.21 void User::setPhone ( const QString & *phone* )

[User::setPhone](#) Modify the number of the desktop *phone*

Parameters

<i>phone</i>	Number of the desktop phone
--------------	-----------------------------

5.53.3.22 void User::setPostalCode ( const QString & *postalCode* )

[User::setPostalCode](#) Modify the postal code *postalCode*

Parameters

<i>postalCode</i>	New postal code
-------------------	-----------------

5.53.3.23 void User::setTitle ( const QString & *title* )

[User::setTitle](#) Modify the user/company activities *description*

Parameters

<i>title</i>	Short description on activity(ies) of customer company
--------------	--

The documentation for this class was generated from the following files:

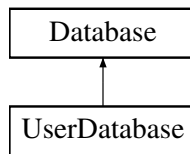
- src/models/user.h
- src/models/user.cpp

## 5.54 UserDatabase Class Reference

The [UserDatabase](#) class Access to [User](#) data in the the table [User](#) of the [Database](#)

```
#include <userdatabase.h>
```

Inheritance diagram for UserDatabase:



## Public Member Functions

- `QStandardItemModel * getUserTable ()` throw (DbException\*)  
*getUserTable* Return an item model of [User](#) for *QTableView*
- `User * getUser (const int pld=1)`  
*getUser* Get informations about the user (identified by 'pld')
- `void updateUser (const User &)`  
*updateUser* Update informations about the user

## Static Public Member Functions

- `static UserDatabase * instance ()` throw (DbException\*)  
*UserDatabase::getInstance* Return an instance of [UserDatabase](#).

## Additional Inherited Members

### 5.54.1 Detailed Description

The [UserDatabase](#) class Access to [User](#) data in the the table [User](#) of the [Database](#)

#### Author

Florent Berbie

#### See also

[Database](#)  
[User](#)

### 5.54.2 Member Function Documentation

#### 5.54.2.1 `User * UserDatabase::getUser ( const int pld = 1 )`

*getUser* Get informations about the user (identified by 'pld')

#### Parameters

<i>pld</i>	user id (1 default) *
------------	-----------------------

#### Returns

the user

#### 5.54.2.2 `QStandardItemModel* UserDatabase::getUserTable ( ) throw DbException *`

`getUserTable` Return an item model of [User](#) for `QTableView`

##### Returns

`QStandardItemModel` an item model

#### 5.54.2.3 `UserDatabase * UserDatabase::instance ( ) throw DbException *` `[static]`

`UserDatabase::getInstance` Return an instance of [UserDatabase](#).

##### Returns

Instance of [UserDatabase](#)

The documentation for this class was generated from the following files:

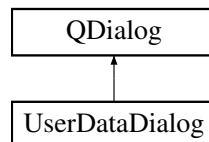
- `src/database/userdatabase.h`
- `src/database/userdatabase.cpp`

## 5.55 UserDataDialog Class Reference

The [UserDataDialog](#) class Window to fill user data.

```
#include <userdatadialog.h>
```

Inheritance diagram for `UserDataDialog`:



### Public Slots

- void [checkFields](#) ()  
*[UserDataDialog::checkFields](#) Check all fields of dialog components.*

### Public Member Functions

- [UserDataDialog](#) (`QWidget *parent=0`)  
*[UserDataDialog](#): Construct a window with user data.*
- void [fillFields](#) ()  
*[UserDataDialog::fillFields](#) Fill line edits with the data of the user.*
- void [accept](#) ()  
*[accept](#) Valid data inputed by user and add these data in [Database](#)*
- void [reject](#) ()  
*[reject](#) Cancel the operation and close the windows*



### 5.55.1 Detailed Description

The [UserDataDialog](#) class Window to fill user data.

Author

Florent Berbie

See also

[Project](#)

### 5.55.2 Constructor & Destructor Documentation

5.55.2.1 `UserDataDialog::UserDataDialog ( QWidget * parent = 0 ) [explicit]`

[UserDataDialog](#): Construct a window with user data.

Parameters

<i>parent</i>	
---------------	--

The documentation for this class was generated from the following files:

- `src/dialogs/userdatadialog.h`
- `src/dialogs/userdatadialog.cpp`

## 5.56 Utils Class Reference

The [Utils](#) class.

```
#include <utils.h>
```

### Static Public Member Functions

- static QString [firstLetterToUpper](#) (QString s)  
*firstLetterToUpper* Put the first letter of a string in capslock

### 5.56.1 Detailed Description

The [Utils](#) class.

Author

Antoine de Roquemaurel

### 5.56.2 Member Function Documentation

5.56.2.1 `QString Utils::firstLetterToUpper ( QString s ) [static]`

`firstLetterToUpper` Put the first letter of a string in capslock

**Parameters**

<i>s</i>	The string to display
----------	-----------------------

**Returns**

The new string with caps

The documentation for this class was generated from the following files:

- src/utls.h
- src/utls.cpp

# Index

- Billing, [15](#)
  - Billing, [16](#)
  - hydrat, [17](#)
  - operator!=, [18](#)
  - operator==, [18](#)
- Contributory, [42](#)
  - Contributory, [43](#)
  - hydrat, [43](#)
- Customer, [46](#)
  - Customer, [48](#)
  - hydrat, [50](#)
  - operator!=, [50](#)
  - operator==, [50](#)
- Database, [57](#)
  - instance, [59](#)
  - value, [59](#)
- hydrat
  - Billing, [17](#)
  - Contributory, [43](#)
  - Customer, [50](#)
  - Project, [76](#)
  - User, [94](#)
- instance
  - Database, [59](#)
  - Log, [67](#)
- Log, [67](#)
  - instance, [67](#)
  - operator<<, [68](#)
  - write, [67](#)
- operator!=
  - Billing, [18](#)
  - Customer, [50](#)
- operator<<
  - Log, [68](#)
- operator==
  - Billing, [18](#)
  - Customer, [50](#)
- Parameters, [73](#)
- Popup, [74](#)
- Project, [74](#)
  - hydrat, [76](#)
  - Project, [75](#)
- Search, [86](#)
- testadder, [91](#)
- User, [91](#)
  - hydrat, [94](#)
  - User, [93](#)
- Utils, [99](#)
- value
  - Database, [59](#)
- write
  - Log, [67](#)