

EasyTicket

Generated by Doxygen 1.9.1

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	5
3.1 Admin Class Reference	5
3.1.1 Constructor & Destructor Documentation	5
3.1.1.1 Admin()	5
3.1.2 Member Function Documentation	6
3.1.2.1 estUnAdmin()	6
3.2 Categorie Class Reference	6
3.2.1 Constructor & Destructor Documentation	6
3.2.1.1 Categorie()	6
3.3 Client Class Reference	7
3.3.1 Constructor & Destructor Documentation	7
3.3.1.1 Client()	7
3.3.2 Member Function Documentation	8
3.3.2.1 estUnClient()	8
3.4 Ingenieur Class Reference	8
3.4.1 Constructor & Destructor Documentation	9
3.4.1.1 Ingenieur()	9
3.4.2 Member Function Documentation	9
3.4.2.1 estUnIngenieur()	9
3.5 Logiciel Class Reference	10
3.5.1 Constructor & Destructor Documentation	10
3.5.1.1 Logiciel()	10
3.6 MainWindow Class Reference	10
3.7 Message Class Reference	11
3.7.1 Member Function Documentation	11
3.7.1.1 getDate_envoie()	11
3.7.1.2 getId_message()	11
3.7.1.3 getTicket()	12
3.7.1.4 getUser()	12
3.7.1.5 setDate_envoie()	12
3.7.1.6 setId_message()	12
3.7.1.7 setTicket()	13
3.7.1.8 setUser()	13
3.8 Personnel Class Reference	13
3.8.1 Constructor & Destructor Documentation	14
3.8.1.1 Personnel()	14
3.8.2 Member Function Documentation	14

3.8.2.1 estUnIngenieur()	14
3.8.2.2 estUnTechnicien()	15
3.9 Systeme Class Reference	15
3.9.1 Constructor & Destructor Documentation	15
3.9.1.1 Systeme()	15
3.9.2 Member Function Documentation	16
3.9.2.1 getId_systeme()	16
3.9.2.2 getNom()	16
3.9.2.3 setId_systeme()	16
3.9.2.4 setNom()	16
3.10 Technicien Class Reference	17
3.10.1 Constructor & Destructor Documentation	17
3.10.1.1 Technicien()	17
3.11 Ticket Class Reference	18
3.11.1 Constructor & Destructor Documentation	19
3.11.1.1 Ticket()	19
3.11.2 Member Function Documentation	19
3.11.2.1 getClient()	19
3.11.2.2 getDate_creation()	19
3.11.2.3 getDate_fermeture()	20
3.11.2.4 getId_ticket()	20
3.11.2.5 getListeMessages()	20
3.11.2.6 getLogiciel()	20
3.11.2.7 getPersonnel()	21
3.11.2.8 getSysteme()	21
3.11.2.9 setClient()	21
3.11.2.10 setDate_creation()	21
3.11.2.11 setDate_fermeture()	22
3.11.2.12 setId_ticket()	22
3.11.2.13 setListeMessages()	22
3.11.2.14 setLogiciel()	22
3.11.2.15 setPersonnel()	23
3.11.2.16 setSysteme()	23
3.12 Utilisateur Class Reference	23
3.12.1 Detailed Description	25
3.12.2 Constructor & Destructor Documentation	25
3.12.2.1 Utilisateur()	25
3.12.3 Member Function Documentation	25
3.12.3.1 estUnAdmin()	25
3.12.3.2 estUnClient()	26
3.12.3.3 estUnPersonnel()	26
3.12.3.4 getId_users()	26

3.12.3.5 getMail()	26
3.12.3.6 getMdp()	27
3.12.3.7 getNom()	27
3.12.3.8 getPrenom()	27
3.12.3.9 setId_users()	27
3.12.3.10 setMail()	28
3.12.3.11 setMdp()	28
3.12.3.12 setNom()	28
3.12.3.13 setPrenom()	28

Index	31
--------------	-----------

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Categorie	6
Logiciel	10
Message	11
QMainWindow	
MainWindow	10
Systeme	15
Ticket	18
Utilisateur	23
Admin	5
Client	7
Personnel	13
Ingenieur	8
Technicien	17

Chapter 2

Class Index

2.1 Class List

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

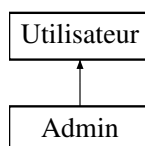
Admin	5
Categorie	6
Client	7
Ingenieur	8
Logiciel	10
MainWindow	10
Message	11
Personnel	13
Systeme	15
Technicien	17
Ticket	18
Utilisateur	
La classe Utilisateur est la classe mère de toutes les implémentations d'utilisateurs possibles	23

Chapter 3

Class Documentation

3.1 Admin Class Reference

Inheritance diagram for Admin:



Public Member Functions

- [Admin](#) ()
La classe [Admin](#) permet de décrire un Administrateur dans l'application EasyTicket.
- **Admin** (std::string [id_user](#), std::string [nom](#), std::string [prenom](#), std::string [mdp](#), std::string id, std::string [mail](#))
- bool [estUnAdmin](#) ()
Cette méthode permet de savoir si un utilisateur est un [Admin](#) ou non.

Additional Inherited Members

3.1.1 Constructor & Destructor Documentation

3.1.1.1 Admin()

```
Admin::Admin ( )
```

La classe [Admin](#) permet de décrire un Administrateur dans l'application EasyTicket.

Parameters

<i>id_user</i>	l'identifiant de l'administrateur
<i>nom</i>	le nom de l'administrateur
<i>prenom</i>	le prenom de l'administrateur
<i>mail</i>	le mail de l'administrateur

3.1.2 Member Function Documentation

3.1.2.1 estUnAdmin()

```
bool Admin::estUnAdmin ( )
```

Cette méthode permet de savoir si un utilisateur est un [Admin](#) ou non.

Returns

true

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

- include/admin.h
- src/admin.cpp

3.2 Categorie Class Reference

Public Member Functions

- [Categorie](#) (std::string id_categorie, std::string nom_categorie)
la classe [Categorie](#) permet de décrire une catégorie de ticket dans l'application EasyTicket.

3.2.1 Constructor & Destructor Documentation

3.2.1.1 Categorie()

```
Categorie::Categorie (
    std::string id_categorie,
    std::string nom_categorie )
```

la classe [Categorie](#) permet de décrire une catégorie de ticket dans l'application EasyTicket.

Parameters

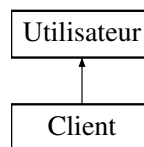
<i>id_categorie</i>	l'identifiant de la catégorie
<i>nom_categorie</i>	le nom de la catégorie

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

- include/categorie.h
- src/categorie.cpp

3.3 Client Class Reference

Inheritance diagram for Client:



Public Member Functions

- [Client](#) (std::string id_u, std::string n, std::string p, std::string motdepasse, std::string email)
La classe [Client](#) permet de décrire un [Client](#) dans l'application EasyTicket.
- bool [estUnClient](#) ()
Cette méthode permet de vérifier si un [Utilisateur](#) est un [Client](#).

Additional Inherited Members

3.3.1 Constructor & Destructor Documentation

3.3.1.1 Client()

```
Client::Client (
    std::string id_u,
    std::string n,
    std::string p,
    std::string motdepasse,
    std::string email )
```

La classe [Client](#) permet de décrire un [Client](#) dans l'application EasyTicket.

Parameters

<i>id_u</i>	L'identifiant de l'utilisateur
<i>n</i>	Le nom de l'utilisateur
<i>p</i>	Le prénom de l'utilisateur
<i>motdepasse</i>	Le mot de passe de l'utilisateur
<i>email</i>	Le mail de l'utilisateur

3.3.2 Member Function Documentation

3.3.2.1 estUnClient()

```
bool Client::estUnClient ( )
```

Cette méthode permet de vérifier si un [Utilisateur](#) est un [Client](#).

Returns

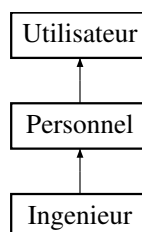
true

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

- include/client.h
- src/client.cpp

3.4 Ingenieur Class Reference

Inheritance diagram for Ingenieur:

**Public Member Functions**

- [Ingenieur](#) (std::string id_u, std::string n, std::string p, std::string motdepasse, std::string email)

La classe [Ingenieur](#) permet de décrire un ingénieur dans l'application EasyTicket.

- bool [estUnIngenieur](#) ()

Cette méthode permet de vérifier si un [Utilisateur](#) est un Ingénieur.

Additional Inherited Members

3.4.1 Constructor & Destructor Documentation

3.4.1.1 Ingenieur()

```
Ingenieur::Ingenieur (
    std::string id_u,
    std::string n,
    std::string p,
    std::string motdepasse,
    std::string email )
```

La classe [Ingenieur](#) permet de décrire un ingénieur dans l'application EasyTicket.

Parameters

<i>id_u</i>	l'identifiant de l'ingénieur
<i>n</i>	le nom de l'ingénieur
<i>p</i>	le prénom de l'ingénieur
<i>motdepasse</i>	le mot de passe de l'ingénieur
<i>email</i>	l'email de l'ingénieur

3.4.2 Member Function Documentation

3.4.2.1 estUnIngenieur()

```
bool Ingenieur::estUnIngenieur ( )
```

Cette méthode permet de vérifier si un [Utilisateur](#) est un Ingénieur.

Returns

true

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

- include/ingenieur.h
- src/ingenieur.cpp

3.5 Logiciel Class Reference

Public Member Functions

- [Logiciel](#) (std::string id_logiciel, std::string nom)

La classe logiciel permet de décrire un logiciel dans l'application EasyTicket.

3.5.1 Constructor & Destructor Documentation

3.5.1.1 Logiciel()

```
Logiciel::Logiciel (
    std::string id_logiciel,
    std::string nom )
```

La classe logiciel permet de décrire un logiciel dans l'application EasyTicket.

Parameters

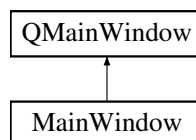
<i>id_logiciel</i>	l'identifiant du logiciel
<i>nom</i>	le nom du logiciel

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

- include/logiciel.h
- src/logiciel.cpp

3.6 MainWindow Class Reference

Inheritance diagram for MainWindow:



Public Member Functions

- **MainWindow** (QWidget *parent=nullptr)

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

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

3.7 Message Class Reference

Public Member Functions

- [Message](#) (std::string id_message, double date_envoie, [Utilisateur](#) user, [Ticket](#) ticket)
Le constructeur par défaut.
- std::string [getId_message](#) () const
Cette méthode permet de récupérer l'identifiant du message.
- void [setId_message](#) (const std::string &value)
Cette méthode permet de définir l'identifiant du message.
- double [getDate_envoie](#) () const
Cette méthode permet de récupérer la date à laquelle le message a été envoyé.
- void [setDate_envoie](#) (double value)
Cette méthode permet de définir la date à laquelle a été envoyé le message.
- [Utilisateur](#) * [getUser](#) () const
Cette méthode permet de récupérer l' Users qui est l'auteur du message.
- void [setUser](#) ([Utilisateur](#) *&value)
Cette méthode permet de définir l' Utilisateur qui l'auteur du message.
- [Ticket](#) * [getTicket](#) () const
Cette méthode permet de récupérer le Ticket dans lequel se trouve le message.
- void [setTicket](#) ([Ticket](#) *&value)
Cette méthode définit le Ticket dans lequel se trouve le message.

3.7.1 Member Function Documentation

3.7.1.1 getDate_envoie()

```
double Message::getDate_envoie ( ) const
```

Cette méthode permet de récupérer la date à laquelle le message a été envoyé.

Returns

la date d'envoi du message

3.7.1.2 getId_message()

```
std::string Message::getId_message ( ) const
```

Cette méthode permet de récupérer l'identifiant du message.

Returns

l'identifiant du message

3.7.1.3 getTicket()

```
Ticket * Message::getTicket ( ) const
```

Cette méthode permet de récupérer le [Ticket](#) dans lequel se trouve le message.

Returns

la classe Users

3.7.1.4 getUser()

```
Utilisateur * Message::getUser ( ) const
```

Cette méthode permet de récupérer l' Users qui est l'auteur du message.

Parameters

<i>value</i>	le nouveau Users
--------------	------------------

3.7.1.5 setDate_envoie()

```
void Message::setDate_envoie (
    double value )
```

Cette méthode permet de définir la date à laquelle a été envoyé le message.

Parameters

<i>value</i>	la nouvelle date d'envoi
--------------	--------------------------

3.7.1.6 setId_message()

```
void Message::setId_message (
    const std::string & value )
```

Cette méthode permet de définir l'identifiant du message.

Parameters

<i>value</i>	Le nouvel identifiant du message
--------------	----------------------------------

3.7.1.7 setTicket()

```
void Message::setTicket (
    Ticket *& value )
```

Cette méthode définit le [Ticket](#) dans lequel se trouve le message.

Parameters

<i>value</i>	le nouveau Ticket
--------------	-----------------------------------

3.7.1.8 setUser()

```
void Message::setUser (
    Utilisateur *& value )
```

Cette méthode permet de définir l' [Utilisateur](#) qui l'auteur du message.

Parameters

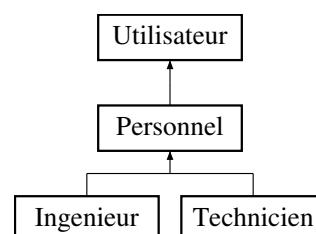
<i>value</i>	le nouveau Users
--------------	------------------

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

- include/message.h
- src/message.cpp

3.8 Personnel Class Reference

Inheritance diagram for Personnel:



Public Member Functions

- [Personnel](#) (std::string id_u, std::string n, std::string p, std::string motdepasse, std::string email)
La classe [Personnel](#) permet de décrire un membre du personnel dans l'application EasyTicket.
- virtual [~Personnel](#) ()=default
Déconstructeur virtuel pour rendre la classe [Personnel](#) abstraite.
- bool [estUnIngenieur](#) ([Personnel](#) *personnel)
Cette méthode permet de vérifier si le [Personnel](#) est un [Personnel](#).
- bool [estUnTechnicien](#) ([Personnel](#) *personnel)
Cette méthode permet de vérifier si un [Personnel](#) est un [Technicien](#).

Additional Inherited Members

3.8.1 Constructor & Destructor Documentation

3.8.1.1 Personnel()

```
Personnel::Personnel (
    std::string id_u,
    std::string n,
    std::string p,
    std::string motdepasse,
    std::string email )
```

La classe [Personnel](#) permet de décrire un membre du personnel dans l'application EasyTicket.

Parameters

<i>id_u</i>	l'identifiant du personnel
<i>n</i>	le nom du personnel
<i>p</i>	le prénom du personnel
<i>email</i>	l'email du personnel

3.8.2 Member Function Documentation

3.8.2.1 estUnIngenieur()

```
bool Personnel::estUnIngenieur (
    Personnel * personnel )
```

Cette méthode permet de vérifier si le [Personnel](#) est un [Personnel](#).

Returns

???

3.8.2.2 estUnTechnicien()

```
bool Personnel::estUnTechnicien (
    Personnel * personnel )
```

Cette méthode permet de vérifier si un [Personnel](#) est un [Technicien](#).

Returns

???

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

- include/personnel.h
- src/personnel.cpp

3.9 Systeme Class Reference

Public Member Functions

- [Systeme](#) (std::string id_systeme, std::string nom)
La classe [Systeme](#) permet de décrire un système d'exploitation dans l'application EasyTicket.
- std::string [getId_systeme](#) () const
Cette méthode permet de récupérer l'identifiant du système.
- void [setId_systeme](#) (const std::string &value)
Cette méthode permet de définir l'identifiant du système.
- std::string [getNom](#) () const
Cette méthode permet de récupérer le nom du système.
- void [setNom](#) (const std::string &value)
Cette méthode permet de définir le nom du système.

3.9.1 Constructor & Destructor Documentation

3.9.1.1 Systeme()

```
Systeme::Systeme (
    std::string id_systeme,
    std::string nom )
```

La classe [Systeme](#) permet de décrire un système d'exploitation dans l'application EasyTicket.

Parameters

<i>id_systeme</i>	L'identifiant du système d'exploitation
<i>nom</i>	Le nom du système d'exploitation

3.9.2 Member Function Documentation

3.9.2.1 getId_systeme()

```
std::string Systeme::getId_systeme ( ) const
```

Cette méthode permet de récupérer l'identifiant du système.

Returns

l'identifiant du système

3.9.2.2 getNom()

```
std::string Systeme::getNom ( ) const
```

Cette méthode permet de récupérer le nom du système.

Returns

le nom du système

3.9.2.3 setId_systeme()

```
void Systeme::setId_systeme (
    const std::string & value )
```

Cette méthode permet de définir l'identifiant du système.

Parameters

<i>value</i>	l'identifiant du système
--------------	--------------------------

3.9.2.4 setNom()

```
void Systeme::setNom (
    const std::string & value )
```

Cette méthode permet de définir le nom du système.

Parameters

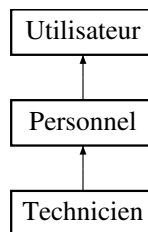
<i>value</i>	le nouveau nom du système
--------------	---------------------------

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

- include/systeme.h
- src/systeme.cpp

3.10 Technicien Class Reference

Inheritance diagram for Technicien:



Public Member Functions

- [Technicien](#) (std::string id_u, std::string n, std::string p, std::string motdepasse, std::string email, std::vector< [Categorie](#) * > liste_categories)

Cette classe permet de décrire un [Technicien](#) dans l'application EasyTicket.

Additional Inherited Members

3.10.1 Constructor & Destructor Documentation

3.10.1.1 Technicien()

```

Technicien::Technicien (
    std::string id_u,
    std::string n,
    std::string p,
    std::string motdepasse,
    std::string email,
    std::vector< Categorie * > liste_categories )

```

Cette classe permet de décrire un [Technicien](#) dans l'application EasyTicket.

Parameters

<i>id_u</i>	l'identifiant du technicien
<i>n</i>	le nom du technicien
<i>p</i>	le prénom du technicien
<i>motdepasse</i>	le mot de passe du technicien
<i>mail</i>	le mail du technicien
<i>liste_categories</i>	la liste des catégories que peut traiter le technicien

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

- include/technicien.h
- src/technicien.cpp

3.11 Ticket Class Reference

Public Member Functions

- [Ticket](#) (std::string id_ticket, double date_creation, [Systeme](#) systeme, [Logiciel](#) logiciel, [Client](#) auteur)
la classe [Ticket](#) permet de décrire un ticket dans l'application EasyTicket.
- std::string [getId_ticket](#) () const
Cette méthode permet de récupérer l'identifiant du [Ticket](#).
- void [setId_ticket](#) (const std::string &value)
Cette méthode permet de définir l'identifiant du [Ticket](#).
- double [getDate_creation](#) () const
Cette méthode permet de récupérer la date de création du ticket.
- void [setDate_creation](#) (double value)
Cette méthode permet de définir la date de création du [Ticket](#).
- double [getDate_fermeture](#) () const
Cette méthode permet de récupérer la date de fermeture du [Ticket](#).
- void [setDate_fermeture](#) (double value)
Cette méthode permet de définir la date de fermeture du [Ticket](#).
- [Systeme](#) [getSysteme](#) () const
Cette méthode permet de récupérer le système d'exploitation concerné par le [Ticket](#).
- void [setSysteme](#) (const [Systeme](#) &value)
Cette méthode permet de définir le [Systeme](#) concerné par le [Ticket](#).
- [Logiciel](#) [getLogiciel](#) () const
Cette méthode permet récupérer le [Logiciel](#) concerné par le [Ticket](#).
- void [setLogiciel](#) (const [Logiciel](#) &value)
Cette méthode permet définir le [Logiciel](#) concerné par le [Ticket](#).
- std::vector< [Message](#) * > [getListeMessages](#) () const
Cette méthode permet de récupérer la liste des [Message](#) concerné par le [Ticket](#).
- void [setListeMessages](#) (const std::vector< [Message](#) * > &value)
Cette méthode permet de définir la liste des @Message concerné par le [Ticket](#).
- void [setClient](#) ([Client](#) &value)
Cette méthode permet de définir le [Client](#) auteur du [Ticket](#).
- [Client](#) [getClient](#) ()
Cette méthode permet de récupérer le [Client](#) auteur du [Ticket](#).
- void [setPersonnel](#) ([Personnel](#) &value)
Cette méthode permet de définir le [Personnel](#) qui traite le [Ticket](#).
- [Personnel](#) [getPersonnel](#) ()
Cette méthode permet de récupérer le [Personnel](#) qui traite le [Ticket](#).

3.11.1 Constructor & Destructor Documentation

3.11.1.1 Ticket()

```
Ticket::Ticket (
    std::string id_ticket,
    double date_creation,
    Systeme systeme,
    Logiciel logiciel,
    Client auteur )
```

la classe [Ticket](#) permet de décrire un ticket dans l'application EasyTicket.

Parameters

<i>id_ticket</i>	l'identifiant du ticket
<i>date_creation</i>	la date de création du ticket
<i>systeme</i>	le Systeme concerné par le ticket
<i>logiciel</i>	le Logiciel concerné par le ticket
<i>auteur</i>	le Client auteur du ticket

3.11.2 Member Function Documentation

3.11.2.1 getClient()

```
Client Ticket::getClient ( )
```

Cette méthode permet de récupérer le [Client](#) auteur du [Ticket](#).

Returns

Le [Client](#) auteur du [Ticket](#)

3.11.2.2 getDate_creation()

```
double Ticket::getDate_creation ( ) const
```

Cette méthode permet de récupérer la date de création du ticket.

Returns

la date de création du ticket

3.11.2.3 getDate_fermeture()

```
double Ticket::getDate_fermeture ( ) const
```

Cette méthode permet de récupérer la date de fermeture du [Ticket](#).

Returns

la date de fermeture du ticket

3.11.2.4 getId_ticket()

```
std::string Ticket::getId_ticket ( ) const
```

Cette méthode permet de récupérer l'identifiant du [Ticket](#).

Returns

l'identifiant du ticket

3.11.2.5 getListeMessages()

```
std::vector< Message * > Ticket::getListeMessages ( ) const
```

Cette méthode permet de récupérer la liste des [Message](#) concerné par le [Ticket](#).

Returns

la liste de [Message](#) concerné par le ticket

3.11.2.6 getLogiciel()

```
Logiciel Ticket::getLogiciel ( ) const
```

Cette méthode permet récupérer le [Logiciel](#) concerné par le [Ticket](#).

Returns

le [Logiciel](#) concerné par le [Ticket](#)

3.11.2.7 getPersonnel()

```
Personnel Ticket::getPersonnel ( )
```

Cette méthode permet de récupérer le [Personnel](#) qui traite le [Ticket](#).

Returns

le Personel traitant le [Ticket](#)

3.11.2.8 getSysteme()

```
Systeme Ticket::getSysteme ( ) const
```

Cette méthode permet de récupérer le système d'exploitation concerné par le [Ticket](#).

Returns

la date de fermeture du ticket

3.11.2.9 setClient()

```
void Ticket::setClient (
    Client & value )
```

Cette méthode permet de définir le [Client](#) auteur du [Ticket](#).

Parameters

<i>value</i>	le Client auteur du Ticket
--------------	--

3.11.2.10 setDate_creation()

```
void Ticket::setDate_creation (
    double value )
```

Cette méthode permet de définir la date de création du [Ticket](#).

Parameters

<i>value</i>	la nouvelle date de création du Ticket
--------------	--

3.11.2.11 setDate_fermeture()

```
void Ticket::setDate_fermeture (
    double value )
```

Cette méthode permet de définir la date de fermeture du [Ticket](#).

Parameters

<i>value</i>	la nouvelle date de fermeture du Ticket
--------------	---

3.11.2.12 setId_ticket()

```
void Ticket::setId_ticket (
    const std::string & value )
```

Cette méthode permet de définir l'identifiant du [Ticket](#).

Parameters

<i>value</i>	le nouvel identifiant du ticket
--------------	---------------------------------

3.11.2.13 setListeMessages()

```
void Ticket::setListeMessages (
    const std::vector< Message * > & value )
```

Cette méthode permet de définir la liste des @Message concerné par le [Ticket](#).

Parameters

<i>value</i>	la nouvelle liste de Message
--------------	--

3.11.2.14 setLogiciel()

```
void Ticket::setLogiciel (
    const Logiciel & value )
```

Cette méthode permet définir le [Logiciel](#) concerné par le [Ticket](#).

Parameters

<i>value</i>	le nouveau Logiciel concerné par le Ticket
--------------	--

3.11.2.15 setPersonnel()

```
void Ticket::setPersonnel (
    Personnel & value )
```

Cette méthode permet de définir le [Personnel](#) qui traite le [Ticket](#).

Parameters

<i>value</i>	le nouveau Personnel
--------------	--------------------------------------

3.11.2.16 setSysteme()

```
void Ticket::setSysteme (
    const Systeme & value )
```

Cette méthode permet de définir le [Systeme](#) concerné par le [Ticket](#).

Parameters

<i>value</i>	le nouveau Systeme concerné par le Ticket
--------------	---

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

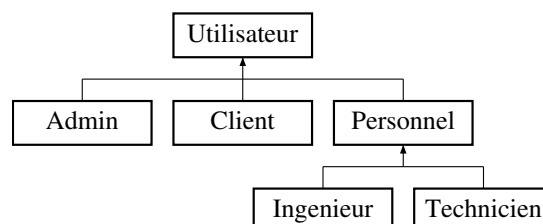
- include/ticket.h
- src/ticket.cpp

3.12 Utilisateur Class Reference

La classe [Utilisateur](#) est la classe mère de toutes les implémentations d'utilisateurs possibles.

```
#include <utilisateur.h>
```

Inheritance diagram for Utilisateur:



Public Member Functions

- [Utilisateur](#) (std::string id_u, std::string n, std::string p, std::string motdepasse, std::string email)
Constructeur avec les données personnelles de l'utilisateur.
- virtual [~Utilisateur](#) ()=default
Déconstructeur virtuel pour rendre la classe.
- std::string [getId_users](#) () const
Cette méthode permet de récupérer l'identifiant de l'utilisateur.
- void [setId_users](#) (const std::string &value)
Cette méthode permet de définir l'identifiant de l'utilisateur.
- std::string [getNom](#) () const
Cette méthode permet de récupérer le nom de l'utilisateur.
- void [setNom](#) (const std::string &value)
Cette méthode permet de définir le nom de l'utilisateur.
- std::string [getPrenom](#) () const
Cette méthode permet de récupérer le prénom de l'utilisateur.
- void [setPrenom](#) (const std::string &value)
Cette méthode permet de définir le prénom de l'utilisateur.
- std::string [getMdp](#) () const
Cette méthode permet de récupérer le mot de passe de l'utilisateur.
- void [setMdp](#) (const std::string &value)
Cette méthode permet de définir le mot de passe de l'utilisateur.
- std::string [getMail](#) () const
Cette méthode permet de récupérer le mail de l'utilisateur.
- void [setMail](#) (const std::string &value)
Cette méthode permet de définir le mail de l'utilisateur.
- bool [estUnClient](#) ([Utilisateur](#) *utilisateur)
Cette méthode permet de vérifier si un [Utilisateur](#) est un client.
- bool [estUnPersonnel](#) ([Utilisateur](#) *utilisateur)
Cette méthode permet de vérifier si le [Utilisateur](#) est un [Personnel](#).
- bool [estUnAdmin](#) ([Utilisateur](#) *utilisateur)
Cette méthode permet de vérifier si un [Utilisateur](#) est un admin.

Protected Attributes

- std::string [id_user](#)
Identifiant de l'utilisateur.
- std::string [nom](#)
Nom de l'utilisateur.
- std::string [prenom](#)
Prénom de l'utilisateur.
- std::string [mdp](#)
Mot de passe de l'utilisateur.
- std::string [mail](#)
E-mail de l'utilisateur.
- std::vector< [Ticket](#) * > [liste_tickets](#)
Liste des tickets créés par l'utilisateur.

3.12.1 Detailed Description

La classe [Utilisateur](#) est la classe mère de toutes les implémentations d'utilisateurs possibles.

3.12.2 Constructor & Destructor Documentation

3.12.2.1 Utilisateur()

```
Utilisateur::Utilisateur (
    std::string id_u,
    std::string n,
    std::string p,
    std::string motdepasse,
    std::string email )
```

Constructeur avec les données personnelles de l'utilisateur.

Parameters

<i>id_u</i>	L'identifiant de l'utilisateur
<i>n</i>	Le nom de l'utilisateur
<i>p</i>	Le prénom de l'utilisateur
<i>motdepasse</i>	Le mot de passe de l'utilisateur
<i>email</i>	Le mail de l'utilisateur

3.12.3 Member Function Documentation

3.12.3.1 estUnAdmin()

```
bool Utilisateur::estUnAdmin (
    Utilisateur * utilisateur )
```

Cette méthode permet de vérifier si un [Utilisateur](#) est un admin.

Returns

???

3.12.3.2 `estUnClient()`

```
bool Utilisateur::estUnClient (
    Utilisateur * utilisateur )
```

Cette méthode permet de vérifier si un `Utilisateur` est un client.

Returns

???

3.12.3.3 `estUnPersonnel()`

```
bool Utilisateur::estUnPersonnel (
    Utilisateur * utilisateur )
```

Cette méthode permet de vérifier si le `Utilisateur` est un `Personnel`.

Returns

???

3.12.3.4 `getId_users()`

```
std::string Utilisateur::getId_users ( ) const
```

Cette méthode permet de récupérer l'identifiant de l'utilisateur.

Returns

L'identifiant de l'utilisateur.

3.12.3.5 `getMail()`

```
std::string Utilisateur::getMail ( ) const
```

Cette méthode permet de récupérer le mail de l'utilisateur.

Returns

Le mail de l'utilisateur.

3.12.3.6 getMdp()

```
std::string Utilisateur::getMdp ( ) const
```

Cette méthode permet de récupérer le mot de passe de l'utilisateur.

Returns

Le mot de passe de l'utilisateur.

3.12.3.7 getNom()

```
std::string Utilisateur::getNom ( ) const
```

Cette méthode permet de récupérer le nom de l'utilisateur.

Returns

Le nom de l'utilisateur.

3.12.3.8 getPrenom()

```
std::string Utilisateur::getPrenom ( ) const
```

Cette méthode permet de récupérer le prénom de l'utilisateur.

Returns

Le prénom de l'utilisateur.

3.12.3.9 setId_users()

```
void Utilisateur::setId_users (
    const std::string & value )
```

Cette méthode permet de définir l'identifiant de l'utilisateur.

Parameters

<i>value</i>	le nouvel identifiant de l'utilisateur.
--------------	---

3.12.3.10 setMail()

```
void Utilisateur::setMail (
    const std::string & value )
```

Cette méthode permet de définir le mail de l'utilisateur.

Parameters

<i>value</i>	le mail
--------------	---------

3.12.3.11 setMdp()

```
void Utilisateur::setMdp (
    const std::string & value )
```

Cette méthode permet de définir le mot de passe de l'utilisateur.

Parameters

<i>value</i>	Le nouveau mot de passe de l'utilisateur.
--------------	---

3.12.3.12 setNom()

```
void Utilisateur::setNom (
    const std::string & value )
```

Cette méthode permet de définir le nom de l'utilisateur.

Parameters

<i>value</i>	le nouveau nom de l'utilisateur.
--------------	----------------------------------

3.12.3.13 setPrenom()

```
void Utilisateur::setPrenom (
    const std::string & value )
```

Cette méthode permet de définir le prénom de l'utilisateur.

Parameters

<i>value</i>	Le nouveau prénom de l'utilisateur.
--------------	-------------------------------------

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

- include/utilisateur.h
- src/utilisateur.cpp

Index

Admin, [5](#)
 Admin, [5](#)
 estUnAdmin, [6](#)

Categorie, [6](#)
 Categorie, [6](#)

Client, [7](#)
 Client, [7](#)
 estUnClient, [8](#)

estUnAdmin
 Admin, [6](#)
 Utilisateur, [25](#)

estUnClient
 Client, [8](#)
 Utilisateur, [25](#)

estUnIngenieur
 Ingenieur, [9](#)
 Personnel, [14](#)

estUnPersonnel
 Utilisateur, [26](#)

estUnTechnicien
 Personnel, [14](#)

getClient
 Ticket, [19](#)

getDate_creation
 Ticket, [19](#)

getDate_envoie
 Message, [11](#)

getDate_fermeture
 Ticket, [19](#)

getId_message
 Message, [11](#)

getId_systeme
 Systeme, [16](#)

getId_ticket
 Ticket, [20](#)

getId_users
 Utilisateur, [26](#)

getListeMessages
 Ticket, [20](#)

getLogiciel
 Ticket, [20](#)

getMail
 Utilisateur, [26](#)

getMdp
 Utilisateur, [26](#)

getNom
 Systeme, [16](#)
 Utilisateur, [27](#)

getPersonnel
 Ticket, [20](#)

getPrenom
 Utilisateur, [27](#)

getSysteme
 Ticket, [21](#)

getTicket
 Message, [11](#)

getUser
 Message, [12](#)

Ingenieur, [8](#)
 estUnIngenieur, [9](#)
 Ingenieur, [9](#)

Logiciel, [10](#)
 Logiciel, [10](#)

MainWindow, [10](#)

Message, [11](#)
 getDate_envoie, [11](#)
 getId_message, [11](#)
 getTicket, [11](#)
 getUser, [12](#)
 setDate_envoie, [12](#)
 setId_message, [12](#)
 setTicket, [13](#)
 setUser, [13](#)

Personnel, [13](#)
 estUnIngenieur, [14](#)
 estUnTechnicien, [14](#)
 Personnel, [14](#)

setClient
 Ticket, [21](#)

setDate_creation
 Ticket, [21](#)

setDate_envoie
 Message, [12](#)

setDate_fermeture
 Ticket, [22](#)

setId_message
 Message, [12](#)

setId_systeme
 Systeme, [16](#)

setId_ticket
 Ticket, [22](#)

setId_users
 Utilisateur, [27](#)

setListeMessages
 Ticket, [22](#)
setLogiciel
 Ticket, [22](#)
setMail
 Utilisateur, [27](#)
setMdp
 Utilisateur, [28](#)
setNom
 Systeme, [16](#)
 Utilisateur, [28](#)
setPersonnel
 Ticket, [23](#)
setPrenom
 Utilisateur, [28](#)
setSysteme
 Ticket, [23](#)
setTicket
 Message, [13](#)
setUser
 Message, [13](#)
Systeme, [15](#)
 getId_systeme, [16](#)
 getNom, [16](#)
 setId_systeme, [16](#)
 setNom, [16](#)
 Systeme, [15](#)

Technicien, [17](#)
 Technicien, [17](#)
Ticket, [18](#)
 getClient, [19](#)
 getDate_creation, [19](#)
 getDate_fermeture, [19](#)
 getId_ticket, [20](#)
 getListeMessages, [20](#)
 getLogiciel, [20](#)
 getPersonnel, [20](#)
 getSysteme, [21](#)
 setClient, [21](#)
 setDate_creation, [21](#)
 setDate_fermeture, [22](#)
 setId_ticket, [22](#)
 setListeMessages, [22](#)
 setLogiciel, [22](#)
 setPersonnel, [23](#)
 setSysteme, [23](#)
 Ticket, [19](#)

Utilisateur, [23](#)
 estUnAdmin, [25](#)
 estUnClient, [25](#)
 estUnPersonnel, [26](#)
 getId_users, [26](#)
 getMail, [26](#)
 getMdp, [26](#)
 getNom, [27](#)
 getPrenom, [27](#)
 setId_users, [27](#)
 setMail, [27](#)
 setMdp, [28](#)
 setNom, [28](#)
 setPrenom, [28](#)
 Utilisateur, [25](#)