SDIApp

1.0

Generated by Doxygen 1.8.17

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

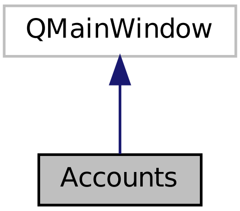
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

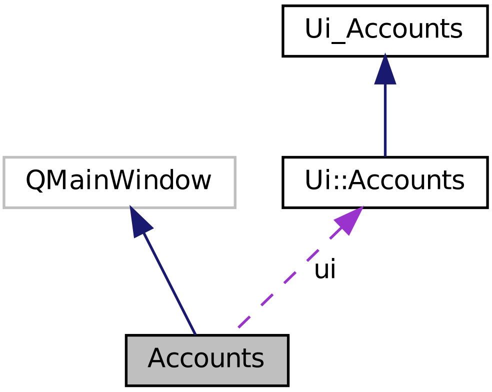
Class Documentation

3.1 Accounts Class Reference

Inheritance diagram for Accounts:



Collaboration diagram for Accounts:



Public Member Functions

- **Accounts** (QWidget ∗parent=nullptr)

Private Slots

- void set_accessibility_settings ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.
- void update_inactivity_status ()

  This function is called every second that the timer adds one to it. Then, inactive_seconds get added by 1 and once they add up to 60, inactive_minutes gets set to 1. Lastly once inactive minutes add up to 15, then both inactive seconds and minutes get reset back to 0 while, the on_logoff_button_clicked function gets called and the user is automatically logged off and directed to the home page. This however, is prevented if at any point the user clicks anywhere on the page.
- void update_date_time ()

  This function updates the current date and time dynamically every second.
- void check_account_type ()

  This function checks if the current_account_type is set to Mod or Admin. If the current_account_type is set to Admin then the main menu button is hidden as the admin must not be able to message other users or join groups. However, if the current_account_type is set to Mod then the main_menu_button is displayed as Mods are able to message other users.
- void load_users ()

  This function loads all the user accounts on the database apart from the admin account into a table, the table shows their profile picture, username and active_status.
- void on_show_users_table_cellClicked (int row, int column)

  This function takes the data from the cell that has been clicked, updates the current_recipient variable and then takes you to the messaging page so that you can message the user that was clicked on.
- void on_home_button_clicked ()

  This function hides the UI and shows the home page.
- void on_chats_button_clicked ()

  This function hides the UI and shows the chats page.
- void on_groups_button_clicked ()

  This function hides the UI and shows the groups page.
- void on_accounts_button_clicked ()

  This function hides the UI and shows the accounts page.
- void on_settings_button_clicked ()

  This function hides the UI and shows the settings page.
- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.
- void on_manage_users_button_clicked ()

  This function hides the UI and shows the admin manage users page.
- void on_manage_groups_button_clicked ()

  This function hides the UI and shows the admin manage groups page.
- void on_manage_mods_button_clicked ()

  This function hides the UI and shows the admin manage mods page.
- void on_logoff_button_clicked ()

  This function is triggered once the user manually clicks the logoff_button or once the inactive_minutes global variable is = 15. An SQL query is run which sets the current users inactive status to 'Offline' in the database whilst redirecting them to the login page.

Private Member Functions

- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.
- void keyPressEvent (QKeyEvent ∗event)

  This function checks for the user pressing keys, if so reset the inactivity settings.

Private Attributes

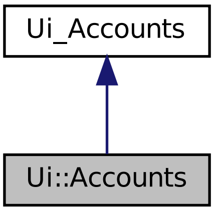
- Ui::Accounts ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

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

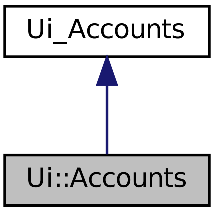
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/accounts.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/accounts.cpp

3.2 Ui::Accounts Class Reference

Inheritance diagram for Ui::Accounts:



Collaboration diagram for Ui::Accounts:



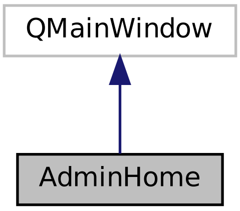
Additional Inherited Members

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

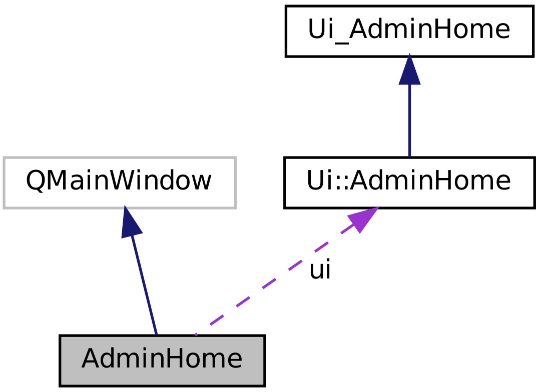
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_accounts.h

3.3 AdminHome Class Reference

Inheritance diagram for AdminHome:



Collaboration diagram for AdminHome:



Public Member Functions

- **AdminHome** (QWidget ∗parent=nullptr)

Private Slots

- void set_accessibility_settings ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.
- void update_inactivity_status ()

  This function is called every second that the timer adds one to it. Then, inactive_seconds get added by 1 and once they add up to 60, inactive_minutes gets set to 1. Lastly once inactive minutes add up to 15, then both inactive seconds and minutes get reset back to 0 while, the on_logoff_button_clicked function gets called and the user is automatically logged off and directed to the home page. This however, is prevented if at any point the user clicks anywhere on the page.
- void on_home_button_clicked ()

  This function is triggered upon clicking the home_button and it redirects the user to the home page. The inactivity↩ _timer is reset as the button has been clicked.
- void on_main_menu_button_clicked ()

  This function is triggered upon clicking the main_menu_button and it redirects the user to the homepage. The inactivity_timer is reset as the button has been clicked and the group manager is set to false to reset the page.
- void on_manage_users_button_clicked ()

  This function is triggered upon clicking the manage_users_button and it redirects the user to the adminManagesUsers page. The inactivity_timer is reset as the button has been clicked.
- void on_manage_groups_button_clicked ()

  This function is triggered upon clicking the manage_groups_button and it redirects the user to the adminManages↩ Groups page. The inactivity_timer is reset as the button has been clicked.
- void on_manage_mods_button_clicked ()

  This function is triggered upon clicking the manage_mods_button and it redirects the user to the adminManagesMods page. The inactivity_timer is reset as the button has been clicked.
- void update_date_time ()

  This function updates the current date and time dynamically every second.
- void check_account_type ()

  This function checks if the current_account_type is set to Mod or Admin. If the current_account_type is set to Admin then the main menu button is hidden as the admin must not be able to message other users or join groups. However, if the current_account_type is set to Mod then the main_menu_button is displayed as Mods are able to message other users.
- void on_logoff_button_clicked ()

  This function is triggered once the user manually clicks the logoff_button or once the inactive_minutes global variable is = 15. An SQL query is run which sets the current users inactive status to 'Offline' in the database whilst redirecting them to the login page.
- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.

Private Member Functions

- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.
- void keyPressEvent (QKeyEvent ∗event)

  This function checks for the user pressing keys, if so reset the inactivity settings.

Private Attributes

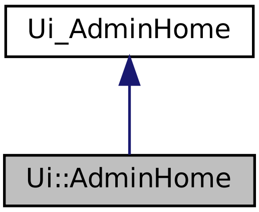
- Ui::AdminHome ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

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

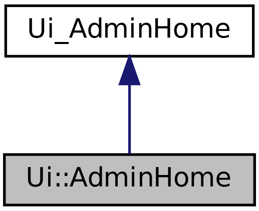
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/adminhome.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/adminhome.cpp

3.4 Ui::AdminHome Class Reference

Inheritance diagram for Ui::AdminHome:



Collaboration diagram for Ui::AdminHome:



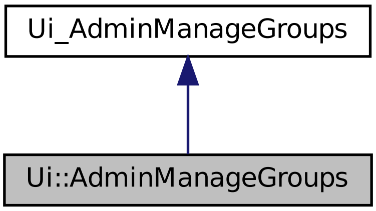
Additional Inherited Members

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

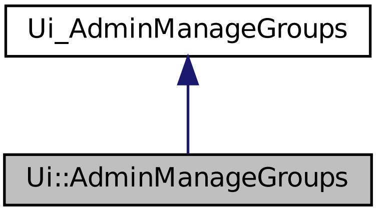
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_adminhome.h

3.5 Ui::AdminManageGroups Class Reference

Inheritance diagram for Ui::AdminManageGroups:



Collaboration diagram for Ui::AdminManageGroups:



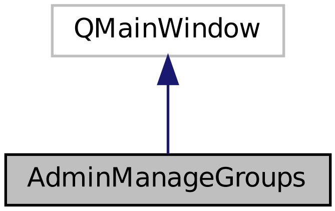
Additional Inherited Members

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

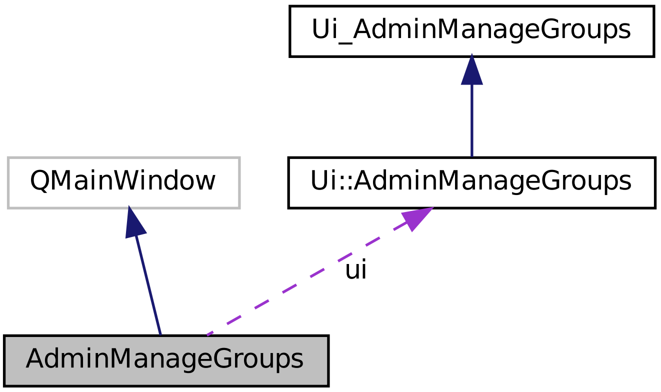
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_adminmanagegroups.h

3.6 AdminManageGroups Class Reference

Inheritance diagram for AdminManageGroups:



Collaboration diagram for AdminManageGroups:



Public Member Functions

- **AdminManageGroups** (QWidget ∗parent=nullptr)

Private Slots

- void update_date_time ()

  This function updates the current date and time dynamically every second.
- void check_account_type ()

  This function checks if the current_account_type is set to Mod or Admin. If the current_account_type is set to Admin then the main menu button is hidden as the admin must not be able to message other users or join groups. However, if the current_account_type is set to Mod then the main_menu_button is displayed as Mods are able to message other users.

- void set_accessibility_settings ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.

- void update_inactivity_status ()

  This function is called every second that the timer adds one to it. Then, inactive_seconds get added by 1 and once they add up to 60, inactive_minutes gets set to 1. Lastly once inactive minutes add up to 15, then both inactive seconds and minutes get reset back to 0 while, the on_logoff_button_clicked function gets called and the user is automatically logged off and directed to the home page. This however, is prevented if at any point the user clicks anywhere on the page.

- void load_one_group ()

  This function simply checks if the group manager is set to true (meaning that they were redirected to this page upon creating the manage groups button on the groups page). It then hides the other admin pages as the group manager (both group adin and group mod) should not be able to view other admin features as they are a group admin and not the system admin.

- void display_one_group ()

  This function displays the table showing information about users within the group that the group manager chose to manage. Only one group is displayed in this function as an admin of one group does not mean that they are the admin of the entire system. First, the database is opened and error messages are shown if there is no connection. Then an SQL query is run to select all the users within the current_recipient's group. The current user and admin are not displayed as you should not be able to promote or demote yourself or the admin. Upon the execution of the query, the users information are updated to the table. This page also has two additional buttons the allow the group manager to promote or demote users within their group.

- void display_all_groups ()

  Similarly to the display_one_group() function, this function displays the all of the groups stored on the database. This function only gets called if the group_manager is set to false meaning that the current_user must be the system admin.

- void on_tableWidget_cellClicked (int row, int column)

  This function first sets inactive_seconds and inactive_time to 0 as the user has been active by selecting a user. It then displays the information of the selected_group_user into the label which then allows the delete_button to set to true so that they can remove the group and the remove_button is shown to allow the group_manager or system admin to remove individual users from the group.

- void on_delete_button_clicked ()

  This function allows the group_manager and the system admin to delete the selected group from the database. The group is deleted from both the groupchat_users and group_chats tables.

- void on_remove_button_clicked ()

  This function allows the group_manager and the system admin to remove individual members from the group(s).

- void on_home_button_clicked ()

  This function is triggered upon clicking the home_button and it redirects the user to the home page. The inactivity←_timer is reset as the button has been clicked.

- void on_manage_users_button_clicked ()

  This function is triggered upon clicking the manage_users_button and it redirects the user to the adminManagesUsers page. The inactivity_timer is reset as the button has been clicked.

- void on_manage_groups_button_clicked ()

  This function is triggered upon clicking the manage_groups_button and it redirects the user to the adminManages←Groups page. The inactivity_timer is reset as the button has been clicked.

- void on_manage_mods_button_clicked ()

  This function is triggered upon clicking the manage_mods_button and it redirects the user to the adminManagesMods page. The inactivity_timer is reset as the button has been clicked.

- void on_main_menu_button_clicked ()

  This function is triggered upon clicking the main_menu_button and it redirects the user to the homepage. The inactivity_timer is reset as the button has been clicked and the group manager is set to false to reset the page.

- void on_logoff_button_clicked ()

  This function is triggered once the user manually clicks the logoff_button or once the inactive_minutes global variable is = 15. An SQL query is run which sets the current users inactive status to 'Offline' in the database whilst redirecting them to the login page.

- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.
- void on_promote_button_clicked ()

  This function allows the group_manager to promote users to moderators of their own group. An SQL query is run to update their user_type to 'Mod'.
- void on_demote_button_clicked ()

  This function allows the group_manager to demote users from being moderators of their group. An SQL query is run to update their user_type to 'Mod'.

Private Member Functions

- void readDBandDisplayTable ()

  checks if the group_manager if false for which it then displays all the groups however, if the group_manager is true it will display only their current group.
- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.
- void keyPressEvent (QKeyEvent ∗event)

  This function checks for the user pressing keys, if so reset the inactivity settings.

Private Attributes

- Ui::AdminManageGroups ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

3.6.1 Member Function Documentation

3.6.1.1 on_tableWidget_cellClicked

```
void AdminManageGroups::on_tableWidget_cellClicked (
            int row,
            int column )  [private], [slot]
```

This function first sets inactive_seconds and inactive_time to 0 as the user has been active by selecting a user. It then displays the information of the selected_group_user into the label which then allows the delete_button to set to true so that they can remove the group and the remove_button is shown to allow the group_manager or system admin to remove individual users from the group.

Parameters

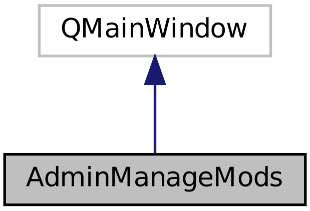
The	row parameter is used to find the information of the user using the rows from the table
The	column parameter is not used.

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

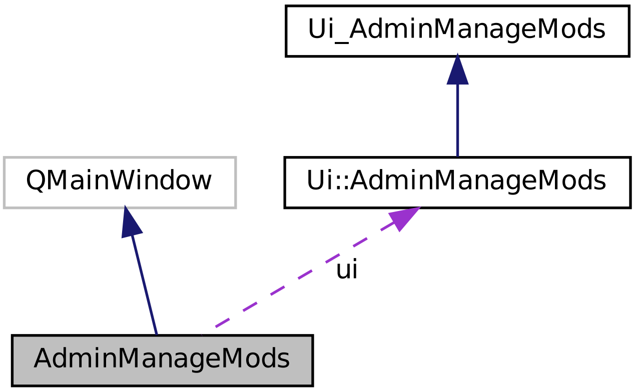
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/adminmanagegroups.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/adminmanagegroups.cpp

3.7 AdminManageMods Class Reference

Inheritance diagram for AdminManageMods:



Collaboration diagram for AdminManageMods:



Public Member Functions

- **AdminManageMods** (QWidget ∗parent=nullptr)

Private Slots

- void set_accessibility_settings ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.

- void update_inactivity_status ()

  This function is called every second that the timer adds one to it. Then, inactive_seconds get added by 1 and once they add up to 60, inactive_minutes gets set to 1. Lastly once inactive minutes add up to 15, then both inactive seconds and minutes get reset back to 0 while, the on_logoff_button_clicked function gets called and the user is automatically logged off and directed to the home page. This however, is prevented if at any point the user clicks anywhere on the page.

- void on_home_button_clicked ()

  This function is triggered upon clicking the home_button and it redirects the user to the home page. The inactivity_timer is reset as the button has been clicked.

- void on_manage_users_button_clicked ()

  This function is triggered upon clicking the manage_users_button and it redirects the user to the adminManagesUsers page. The inactivity_timer is reset as the button has been clicked.

- void on_manage_groups_button_clicked ()

  This function is triggered upon clicking the manage_groups_button and it redirects the user to the adminManagesGroups page. The inactivity_timer is reset as the button has been clicked.

- void on_manage_mods_button_clicked ()

  This function is triggered upon clicking the manage_mods_button and it redirects the user to the adminManagesMods page. The inactivity_timer is reset as the button has been clicked.

- void on_main_menu_button_clicked ()

  This function is triggered upon clicking the main_menu_button and it redirects the user to the homepage. The inactivity_timer is reset as the button has been clicked and the group manager is set to false to reset the page.

- void update_date_time ()

  This function updates the current date and time dynamically every second.

- void check_account_type ()

  This function checks if the current_account_type is set to Mod or Admin. If the current_account_type is set to Admin then the main menu button is hidden as the admin must not be able to message other users or join groups. However, if the current_account_type is set to Mod then the main_menu_button is displayed as Mods are able to message other users.

- void on_tableWidget_cellClicked (int row, int column)

  This function first sets inactive_seconds and inactive_time to 0 as the user has been active by selecting a user. It then displays the information of the selected_mod into the label which then allows the promote_button to set to true so that the system admin can promote users to 'Mod' and the demote_button is shown to allow the system admin to demote users from the system.

- void on_promote_button_clicked ()

  This function allows the system admin to promote the user to 'Mod' of the system. This in turn allows them to manage other groups and users. However, the mod cannot manage the admin.

- void on_demote_button_clicked ()

  This function allows the system admin to demote the user to 'User' of the system. This will then remove Moderator permissions meaning that they cannot manage groups or users anymore.

- void on_logoff_button_clicked ()

  This function is triggered once the user manually clicks the logoff_button or once the inactive_minutes global variable is = 15. An SQL query is run which sets the current users inactive status to 'Offline' in the database whilst redirecting them to the login page.

- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.

Private Member Functions

- void readDBandDisplayTable ()
- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.
- void keyPressEvent (QKeyEvent ∗event)

  This function checks for the user pressing keys, if so reset the inactivity settings.

Private Attributes

- Ui::AdminManageMods ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

3.7.1 Member Function Documentation

3.7.1.1 on_tableWidget_cellClicked

```
void AdminManageMods::on_tableWidget_cellClicked (
             int row,
             int column )  [private], [slot]
```

This function first sets inactive_seconds and inactive_time to 0 as the user has been active by selecting a user. It then displays the information of the selected_mod into the label which then allows the promote_button to set to true so that the system admin can promote users to 'Mod' and the demote_button is shown to allow the system admin to demote users from the system.

Parameters

The	row parameter is used to find the information of the user using the rows from the table
The	column parameter is not used.

3.7.1.2 readDBandDisplayTable()

```
void AdminManageMods::readDBandDisplayTable ( )  [private]
```

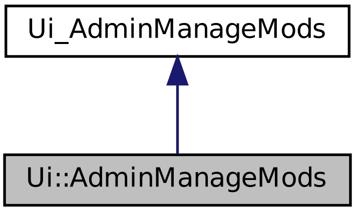
This function displays the table showing information about system moderators. First the inactive_seconds and inactive_minutes get reset back to 0. Then an sql query is run to select the user ID, profile picutre, username and user type and loads it into the table.

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

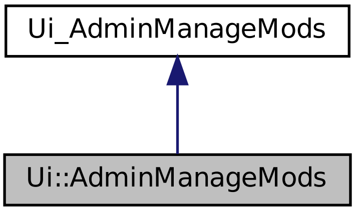
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/adminmanagemods.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/adminmanagemods.cpp

3.8 Ui::AdminManageMods Class Reference

Inheritance diagram for Ui::AdminManageMods:



Collaboration diagram for Ui::AdminManageMods:



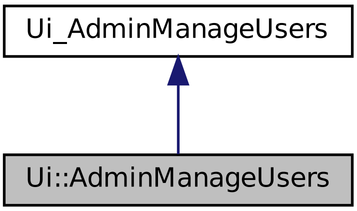
Additional Inherited Members

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

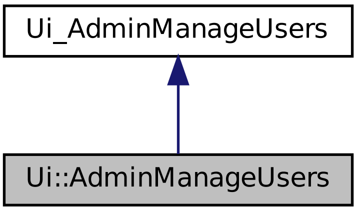
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_adminmanagemods.h

3.9 Ui::AdminManageUsers Class Reference

Inheritance diagram for Ui::AdminManageUsers:



Collaboration diagram for Ui::AdminManageUsers:



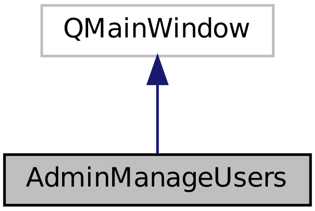
Additional Inherited Members

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

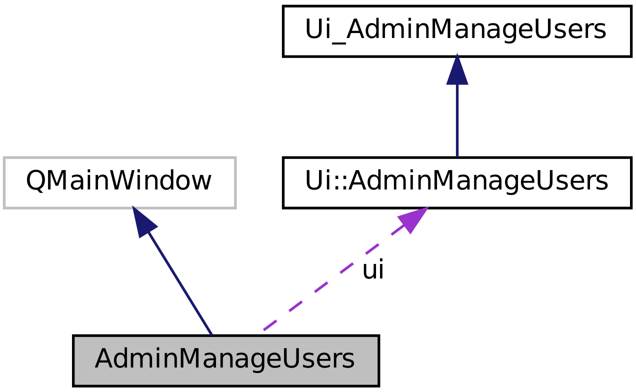
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_adminmanageusers.h

3.10 AdminManageUsers Class Reference

Inheritance diagram for AdminManageUsers:



Collaboration diagram for AdminManageUsers:



Public Member Functions

- **AdminManageUsers** (QWidget ∗parent=nullptr)

Private Slots

- void update_date_time ()

  This function updates the current date and time dynamically every second.
- void set_accessibility_settings ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.

- void update_inactivity_status ()

  This function is called every second that the timer adds one to it. Then, inactive_seconds get added by 1 and once they add up to 60, inactive_minutes gets set to 1. Lastly once inactive minutes add up to 15, then both inactive seconds and minutes get reset back to 0 while, the on_logoff_button_clicked function gets called and the user is automatically logged off and directed to the home page. This however, is prevented if at any point the user clicks anywhere on the page.
- void check_account_type ()

  This function checks if the current_account_type is set to Mod or Admin. If the current_account_type is set to Admin then the main menu button is hidden as the admin must not be able to message other users or join groups. However, if the current_account_type is set to Mod then the main_menu_button is displayed as Mods are able to message other users.
- void on_tableWidget_cellClicked (int row, int column)

  This function first sets inactive_seconds and inactive_time to 0 as the user has been active by selecting a user. The selected_user's information is then displayed in a label using the 'row' parameter to find the values from the table. The delete_button and reset_button are then shown as a user has been selected.
- void on_delete_button_clicked ()

  This function removes the user from the database once they have been selected by the system admin. An sql query is run which removes the user and then calls the function to load the updated user information back into the table.
- void on_reset_button_clicked ()

  This function resets the selected users password. The resetted password is encrypted to 'password'.
- void on_home_button_clicked ()

  This function is triggered upon clicking the home_button and it redirects the user to the home page. The inactivity_timer is reset as the button has been clicked.
- void on_manage_users_button_clicked ()

  This function is triggered upon clicking the manage_users_button and it redirects the user to the adminManagesUsers page. The inactivity_timer is reset as the button has been clicked.
- void on_manage_groups_button_clicked ()

  This function is triggered upon clicking the manage_groups_button and it redirects the user to the adminManagesGroups page. The inactivity_timer is reset as the button has been clicked.
- void on_manage_mods_button_clicked ()

  This function is triggered upon clicking the manage_mods_button and it redirects the user to the adminManagesMods page. The inactivity_timer is reset as the button has been clicked.
- void on_main_menu_button_clicked ()

  This function is triggered upon clicking the main_menu_button and it redirects the user to the homepage. The inactivity_timer is reset as the button has been clicked and the group manager is set to false to reset the page.
- void on_logoff_button_clicked ()

  This function is triggered once the user manually clicks the logoff_button or once the inactive_minutes global variable is = 15. An SQL query is run which sets the current users inactive status to 'Offline' in the database whilst redirecting them to the login page.
- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.

Private Member Functions

- void readDBandDisplayTable ()

  This function displays the table of users from the database. First the inactive_seconds and inactive_minutes get reset back to 0. Then an sql query is run to select the users information and paste them into the styled, interactive table.
- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.
- void keyPressEvent (QKeyEvent ∗event)

  This function checks for the user pressing keys, if so reset the inactivity settings.

Private Attributes

- Ui::AdminManageUsers ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

3.10.1 Member Function Documentation

3.10.1.1 on_tableWidget_cellClicked

```
void AdminManageUsers::on_tableWidget_cellClicked (
             int row,
             int column )  [private], [slot]
```

This function first sets inactive_seconds and inactive_time to 0 as the user has been active by selecting a user. The selected_user's information is then displayed in a label using the 'row' parameter to find the values from the table. The delete_button and reset_button are then shown as a user has been selected.

Parameters

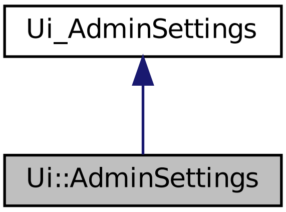
The	row parameter is used to find the information of the user using the rows from the table
The	column parameter is not used.

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

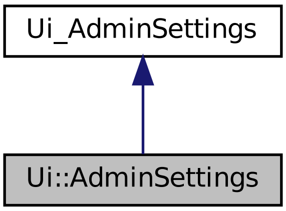
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/adminmanageusers.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/adminmanageusers.cpp

3.11 Ui::AdminSettings Class Reference

Inheritance diagram for Ui::AdminSettings:



Collaboration diagram for Ui::AdminSettings:



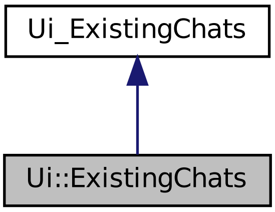
Additional Inherited Members

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

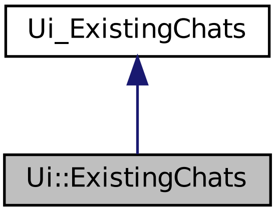
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_adminsettings.h

3.12 Ui::ExistingChats Class Reference

Inheritance diagram for Ui::ExistingChats:



Collaboration diagram for Ui::ExistingChats:



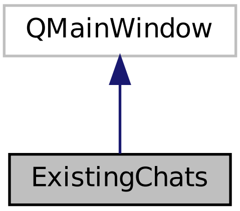
Additional Inherited Members

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

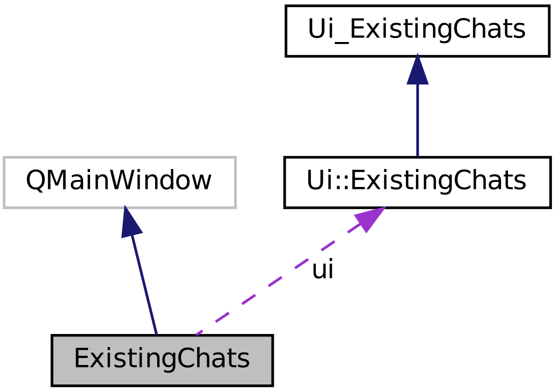
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_existingchats.h

3.13 ExistingChats Class Reference

Inheritance diagram for ExistingChats:



Collaboration diagram for ExistingChats:



Public Member Functions

- **ExistingChats** (QWidget ∗parent=nullptr)

Private Slots

- void set_accessibility_settings ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.
- void update_inactivity_status ()

  This function is called every second that the timer adds one to it. Then, inactive_seconds get added by 1 and once they add up to 60, inactive_minutes gets set to 1. Lastly once inactive minutes add up to 15, then both inactive seconds and minutes get reset back to 0 while, the on_logoff_button_clicked function gets called and the user is automatically logged off and directed to the home page. This however, is prevented if at any point the user clicks anywhere on the page.
- void update_date_time ()

  This function updates the current date and time dynamically every second.
- void check_account_type ()

  This function checks if the current_account_type is set to Mod or Admin. If the current_account_type is set to Admin then the main menu button is hidden as the admin must not be able to message other users or join groups. However, if the current_account_type is set to Mod then the main_menu_button is displayed as Mods are able to message other users.
- void load_existing_chats ()

  This function loads all the user accounts that the user has a chat set up with. This data is loaded into a table which will allow the user to select who they want to message.
- void on_show_existing_chats_table_cellClicked (int row, int column)

  This function takes the data from the cell that has been clicked, updates the current_recipient variable and then takes you to the messaging page so that you can message the user that was clicked on.
- void on_home_button_clicked ()

  This function hides the UI and shows the home page.
- void on_chats_button_clicked ()

This function hides the UI and shows the chats page.

- void on_groups_button_clicked ()

  This function hides the UI and shows the groups page.
- void on_accounts_button_clicked ()

  This function hides the UI and shows the accounts page.
- void on_settings_button_clicked ()

  This function hides the UI and shows the settings page.
- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.
- void on_manage_users_button_clicked ()

  This function hides the UI and shows the admin manage users page.
- void on_manage_groups_button_clicked ()

  This function hides the UI and shows the admin manage groups page.
- void on_manage_mods_button_clicked ()

  This function hides the UI and shows the admin manage mods page.
- void on_logoff_button_clicked ()

  This function is triggered once the user manually clicks the logoff_button or once the inactive_minutes global variable is = 15. An SQL query is run which sets the current users inactive status to 'Offline' in the database whilst redirecting them to the home page.

Private Member Functions

- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.
- void keyPressEvent (QKeyEvent ∗event)

  This function checks for the user pressing keys, if so reset the inactivity settings.

Private Attributes

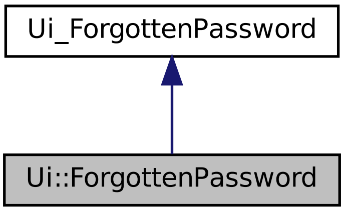
- Ui::ExistingChats ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

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

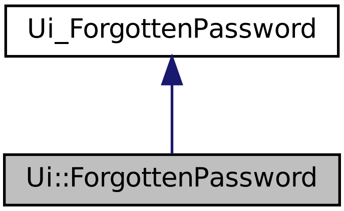
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/existingchats.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/existingchats.cpp

3.14 Ui::ForgottenPassword Class Reference

Inheritance diagram for Ui::ForgottenPassword:



Collaboration diagram for Ui::ForgottenPassword:



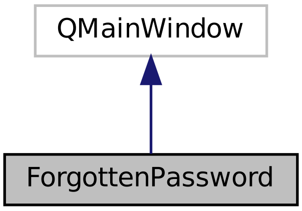
Additional Inherited Members

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

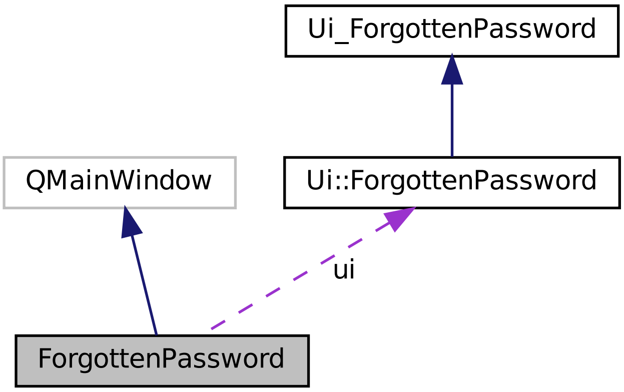
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_forgottenpassword.h

3.15 ForgottenPassword Class Reference

Inheritance diagram for ForgottenPassword:



Collaboration diagram for ForgottenPassword:



Public Member Functions

- **ForgottenPassword** (QWidget ∗parent=nullptr)

Private Slots

- void update_date_time ()

  This function updates the current date and time dynamically every second.
- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.

- void on_change_password_button_clicked ()

  Validates the username and special word entered with the entries in the database, then carries out the password validation. After this the updated password is saved to the database.
- void on_back_to_login_button_clicked ()

  Takes all of the user inputs into each lineedit, validates them by checking all details are filled in, makes sure the username is unique and password has at least 8 characters, special characters and a number. Once these have been confirmed they are saved to the database.
- QString Encrypt (QString password_unencrypted, int key)

  Encrypts the user password using a Caesar cipher.

Private Member Functions

- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.

Private Attributes

- Ui::ForgottenPassword ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

3.15.1 Member Function Documentation

3.15.1.1 Encrypt

```
QString ForgottenPassword::Encrypt (
             QString password_unencrypted,
             int key )  [private], [slot]
```

Encrypts the user password using a Caesar cipher.

Parameters

password_unencrypted	
key	

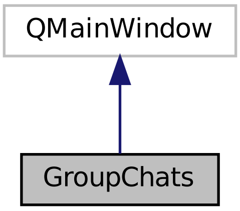
Returns

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

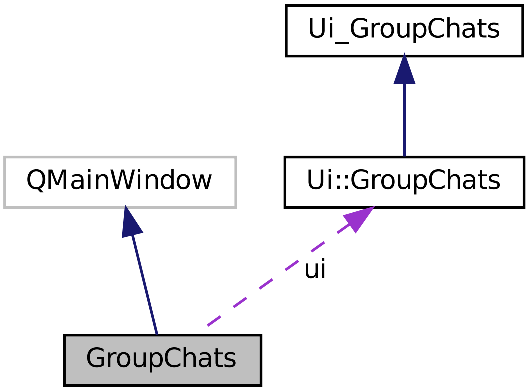
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/forgottenpassword.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/forgottenpassword.cpp

3.16 GroupChats Class Reference

Inheritance diagram for GroupChats:



Collaboration diagram for GroupChats:



Public Member Functions

- **GroupChats** (QWidget ∗parent=nullptr)

Private Slots

- void set_accessibility_settings ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.
- void update_inactivity_status ()

This function is called every second that the timer adds one to it. Then, inactive_seconds get added by 1 and once they add up to 60, inactive_minutes gets set to 1. Lastly once inactive minutes add up to 15, then both inactive seconds and minutes get reset back to 0 while, the on_logoff_button_clicked function gets called and the user is automatically logged off and directed to the home page. This however, is prevented if at any point the user clicks anywhere on the page.

- void update_date_time ()

  This function updates the current date and time dynamically every second.

- void check_account_type ()

  This function checks if the current_account_type is set to Mod or Admin. If the current_account_type is set to Admin then the main menu button is hidden as the admin must not be able to message other users or join groups. However, if the current_account_type is set to Mod then the main_menu_button is displayed as Mods are able to message other users.

- void load_user_accounts ()

  This function loads all the user accounts on the database apart from the admin account into a table, two other columns are added to hold check boxes which allow users to be added into the group as well as also having the option to make them moderators.

- void on_create_group_button_clicked ()

  This function checks the table for the selected checkboxes, if the add useer checkbox is ticked, the linked user is added to the group, if the moderator button is checked, they are added to the group as a moderator, once all the users are added to the group, the user who created the group is added as the group admin. Each time a new user needs to be added, a function is called to add the user to the group in the database.

- void upload_to_database (int group_id, int user_id, QString user_type)

  This function uses the data passed from on_create_button_clicked(); and uploads that data to the database.

- void on_home_button_clicked ()

  This function hides the UI and shows the home page.

- void on_chats_button_clicked ()

  This function hides the UI and shows the chats page.

- void on_groups_button_clicked ()

  This function hides the UI and shows the groups page.

- void on_accounts_button_clicked ()

  This function hides the UI and shows the accounts page.

- void on_settings_button_clicked ()

  This function hides the UI and shows the settings page.

- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.

- void on_manage_users_button_clicked ()

  This function hides the UI and shows the admin manage users page.

- void on_manage_groups_button_clicked ()

  This function hides the UI and shows the admin manage groups page.

- void on_manage_mods_button_clicked ()

  This function hides the UI and shows the admin manage mods page.

- void on_logoff_button_clicked ()

  This function is triggered once the user manually clicks the logoff_button or once the inactive_minutes global variable is = 15. An SQL query is run which sets the current users inactive status to 'Offline' in the database whilst redirecting them to the login page.

- void on_add_remove_user_table_cellClicked (int row, int column)

  This function resets the inactivity settings when a cell is clicked.

- void on_set_group_name_cursorPositionChanged (int arg1, int arg2)

  This function resets the inactivity settings when the cursor position of the input field has changed.

Private Member Functions

- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.
- void keyPressEvent (QKeyEvent ∗event)

  This function checks for the user pressing keys, if so reset the inactivity settings.

Private Attributes

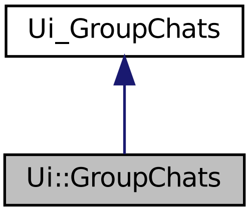
- Ui::GroupChats ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

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

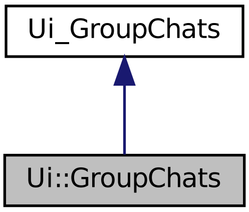
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/groupchats.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/groupchats.cpp

3.17 Ui::GroupChats Class Reference

Inheritance diagram for Ui::GroupChats:



Collaboration diagram for Ui::GroupChats:



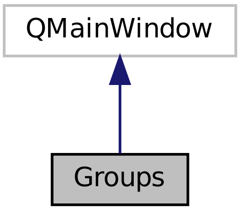
Additional Inherited Members

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

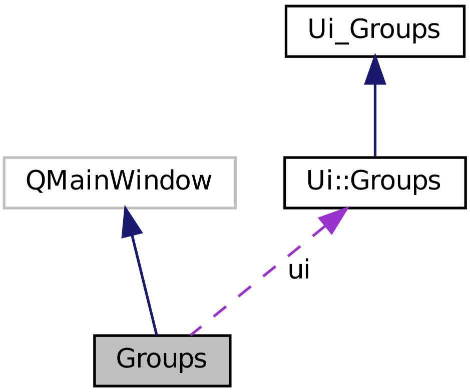
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_groupchats.h

3.18 Groups Class Reference

Inheritance diagram for Groups:



Collaboration diagram for Groups:



Public Member Functions

- **Groups** (QWidget ∗parent=nullptr)

Private Slots

- void set_accessibility_settings ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.
- void update_inactivity_status ()

  This function is called every second that the timer adds one to it. Then, inactive_seconds get added by 1 and once they add up to 60, inactive_minutes gets set to 1. Lastly once inactive minutes add up to 15, then both inactive seconds and minutes get reset back to 0 while, the on_logoff_button_clicked function gets called and the user is automatically logged off and directed to the home page. This however, is prevented if at any point the user clicks anywhere on the page.
- void update_date_time ()

  This function updates the current date and time dynamically every second.
- void check_account_type ()

  This function checks if the current_account_type is set to Mod or Admin. If the current_account_type is set to Admin then the main menu button is hidden as the admin must not be able to message other users or join groups. However, if the current_account_type is set to Mod then the main_menu_button is displayed as Mods are able to message other users.
- void on_create_group_button_clicked ()

  This function hides the UI and shows the create group chat page.
- void load_user_groups ()

  This function loads all the group chats on the database that the user is a member of, the table shows the group chat name.
- void on_show_groupchats_table_cellClicked (int row, int column)

  This function takes the data from the cell that has been clicked, updates the current_recipient variable and then takes you to the messaging page so that you can message the group that was clicked on.
- void on_home_button_clicked ()

  This function hides the UI and shows the home page.
- void on_chats_button_clicked ()

  This function hides the UI and shows the chats page.
- void on_groups_button_clicked ()

  This function hides the UI and shows the groups page.
- void on_accounts_button_clicked ()

  This function hides the UI and shows the accounts page.
- void on_settings_button_clicked ()

  This function hides the UI and shows the settings page.
- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.
- void on_manage_users_button_clicked ()

  This function hides the UI and shows the admin manage users page.
- void on_manage_groups_button_clicked ()

  This function hides the UI and shows the admin manage groups page.
- void on_manage_mods_button_clicked ()

  This function hides the UI and shows the admin manage mods page.
- void on_logoff_button_clicked ()

  This function is triggered once the user manually clicks the logoff_button or once the inactive_minutes global variable is = 15. An SQL query is run which sets the current users inactive status to 'Offline' in the database whilst redirecting them to the login page.

Private Member Functions

- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.
- void keyPressEvent (QKeyEvent ∗event)

  This function checks for the user pressing keys, if so reset the inactivity settings.

Private Attributes

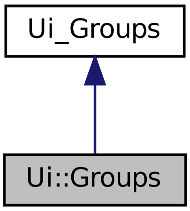
- Ui::Groups ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

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

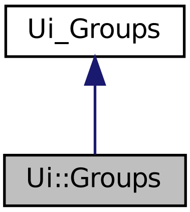
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/groups.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/groups.cpp

3.19 Ui::Groups Class Reference

Inheritance diagram for Ui::Groups:



Collaboration diagram for Ui::Groups:



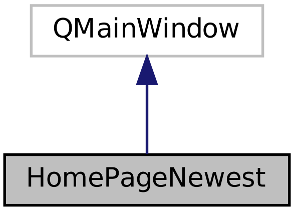
Additional Inherited Members

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

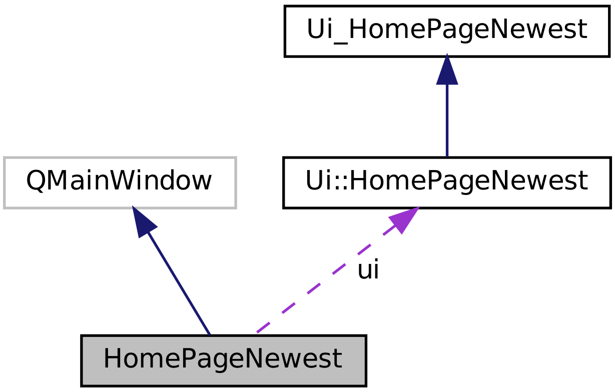
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_groups.h

3.20 HomePageNewest Class Reference

Inheritance diagram for HomePageNewest:



Collaboration diagram for HomePageNewest:



Public Member Functions

- **HomePageNewest** (QWidget ∗parent=nullptr)

Private Slots

- void update_inactivity_status ()

  This function is called every second that the timer adds one to it. Then, inactive_seconds get added by 1 and once they add up to 60, inactive_minutes gets set to 1. Lastly once inactive minutes add up to 15, then both inactive seconds and minutes get reset back to 0 while, the on_logoff_button_clicked function gets called and the user is automatically logged off and directed to the home page. This however, is prevented if at any point the user clicks anywhere on the page.
- void check_account_type ()

  This function checks if the current_account_type is set to Mod or Admin. If the current_account_type is set to Admin then the main menu button is hidden as the admin must not be able to message other users or join groups. However, if the current_account_type is set to Mod then the main_menu_button is displayed as Mods are able to message other users.
- void update_date_time ()

  This function updates the current date and time dynamically every second.
- void on_home_button_clicked ()

  This function hides the UI and shows the home page.
- void on_chats_button_clicked ()

  This function hides the UI and shows the chats page.
- void on_groups_button_clicked ()

  This function hides the UI and shows the groups page.
- void on_accounts_button_clicked ()

  This function hides the UI and shows the accounts page.
- void on_settings_button_clicked ()

  This function hides the UI and shows the settings page.
- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.
- void on_manage_users_button_clicked ()

  This function hides the UI and shows the admin manage users page.
- void on_manage_groups_button_clicked ()

  This function hides the UI and shows the admin manage groups page.
- void on_manage_mods_button_clicked ()

  This function hides the UI and shows the admin manage mods page.
- void on_logoff_button_clicked ()

  This function is triggered once the user manually clicks the logoff_button or once the inactive_minutes global variable is = 15. An SQL query is run which sets the current users inactive status to 'Offline' in the database whilst redirecting them to the login page.
- void set_accessibility_settings ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.

Private Member Functions

- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.
- void keyPressEvent (QKeyEvent ∗event)

  This function checks for the user pressing keys, if so reset the inactivity settings.

Private Attributes

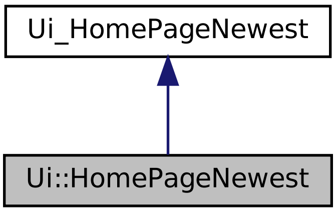
- Ui::HomePageNewest ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

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

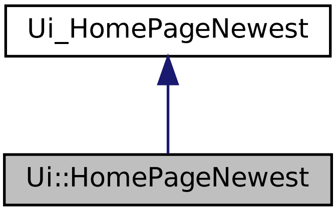
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/homepage.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/homepage.cpp

3.21 Ui::HomePageNewest Class Reference

Inheritance diagram for Ui::HomePageNewest:



Collaboration diagram for Ui::HomePageNewest:



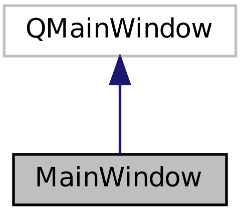
Additional Inherited Members

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

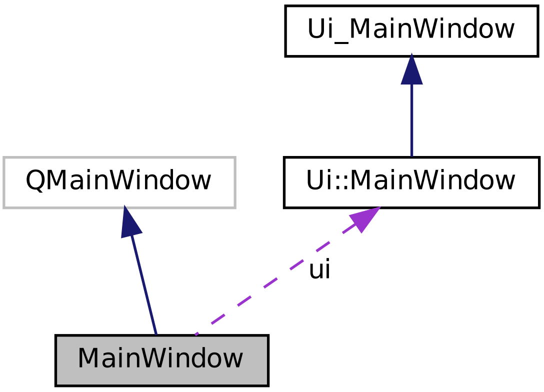
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_homepage.h

3.22 MainWindow Class Reference

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:



Public Member Functions

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

Private Slots

- void on_register_button_clicked ()
- void on_login_button_clicked ()
- void update_date_time ()
- QString EncryptMain (QString password_unencrypted, int key)
- QString DecryptMain (QString password_encrypted, int key)

This is used to decrypted the encryped values, works by putting the encrypted value back into the encrytion function except using the negative encrytpion key of what was used to encrypt the password in the first place. Used for testing purposes.

- void on_forgotten_password_button_clicked ()

  Opens the forgotten password page and hides the login page.
- void load_profile_picture ()

  When the user has successfully logged in this function is called to retrieve the user profile pic from the database, which matches the users username.This image is loaded into the 'image' global variable.
- void on_shutdown_button_clicked ()

  When shutdown button clicked the application is closed.

Private Member Functions

- void mousePressEvent (QMouseEvent ∗event)

  mousePressEvent
- void mouseMoveEvent (QMouseEvent ∗event)

  mouseMoveEvent

Private Attributes

- Ui::MainWindow ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

3.22.1 Member Function Documentation

3.22.1.1 DecryptMain

```
QString MainWindow::DecryptMain (
              QString password_encrypted,
              int key )  [private], [slot]
```

This is used to decrypted the encryped values, works by putting the encrypted value back into the encrytion function except using the negative encrytpion key of what was used to encrypt the password in the first place. Used for testing purposes.

Parameters

password_encrypted	is the encrypted password that needs to be decrypted (testing only)
key	is opposite encryption key used to encrypt the password

Returns

returns password decrypted

3.22.1.2 EncryptMain

```
QString MainWindow::EncryptMain (
             QString password_unencrypted,
             int key )  [private], [slot]
```

\This function uses a caesar cipher to encrypt the entered user password in order to compare the result to the encrypted password in the database.

Parameters

password_unencrypted	is user password entered into the application
key	is the encryption key used to encrypt the password.

Returns

returns the password encrypted

3.22.1.3 mouseMoveEvent()

```
void MainWindow::mouseMoveEvent (
             QMouseEvent * event )  [private]
```

mouseMoveEvent

Parameters

event	

3.22.1.4 mousePressEvent()

```
void MainWindow::mousePressEvent (
             QMouseEvent * event )  [private]
```

mousePressEvent

Parameters

event	

3.22.1.5 on_login_button_clicked

```
void MainWindow::on_login_button_clicked ( )  [private], [slot]
```

\The username and password entered are compared to the existing details in the database, if the details are correct then the home page will be displayed. When this also happens the username is loaded into a gloabal variable called 'current_user' as well as other details to make app features function.

3.22.1.6 on_register_button_clicked

```
void MainWindow::on_register_button_clicked ( )  [private], [slot]
```

\The register page is displayed when this function is run and the login page is hidden.

3.22.1.7 update_date_time

```
void MainWindow::update_date_time ( )  [private], [slot]
```

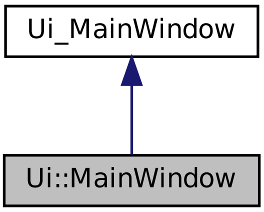
\Update time function simply updates the date and time label to the system date and time.

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

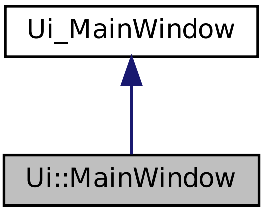
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/mainwindow.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/mainwindow.cpp

3.23 Ui::MainWindow Class Reference

Inheritance diagram for Ui::MainWindow:



Collaboration diagram for Ui::MainWindow:



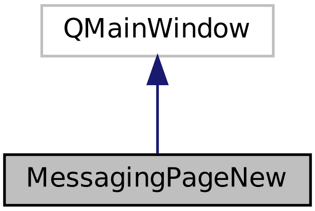
Additional Inherited Members

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

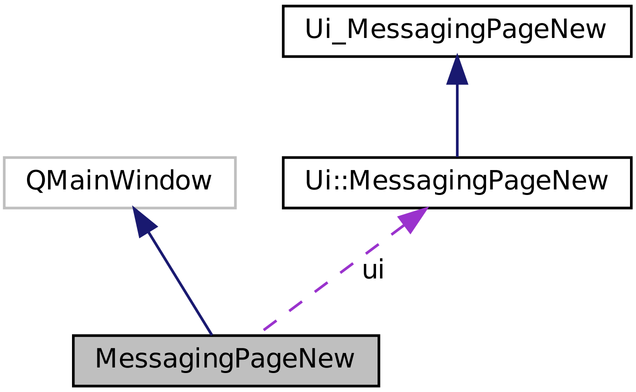
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_mainwindow.h

3.24 MessagingPageNew Class Reference

Inheritance diagram for MessagingPageNew:



Collaboration diagram for MessagingPageNew:



Public Member Functions

- **MessagingPageNew** (QWidget ∗parent=nullptr)

Private Slots

- void set_accessibility_settings ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.
- void update_inactivity_status ()

  This function is called every second that the timer adds one to it. Then, inactive_seconds get added by 1 and once they add up to 60, inactive_minutes gets set to 1. Lastly once inactive minutes add up to 15, then both inactive seconds and minutes get reset back to 0 while, the on_logoff_button_clicked function gets called and the user is automatically logged off and directed to the home page. This however, is prevented if at any point the user clicks anywhere on the page.
- void update_date_time ()

  This function updates the current date and time dynamically every second.
- void search_account_permissions ()

  This function checks the users permissions in a groupchat, if they are a group admin or moderator then they will be given additional permissions, this is done by showing or hiding a button depending on the permissions available to the user.
- void set_active_status ()

  This function checks the active status of the recipient if the user is privately messaging them and displays it to the user. If the recipient is online, then it displays the UI element to 'Online', if they are offline, it displays the UI element to 'Offline'.
- void on_send_button_clicked ()

  This function updates the users UI with the message they want to send and sends it to the recipient. For updating the users UI, the message is styled with a QTextBlockFormat to be left aligned and also styles the block to make message belonging more visible to the user. If the user is messaging one user the message date and time are sent, if they are messaging a groupchat, it sends the message, date and time and their username to show who sent the message in the groupchat.
- void subscribe_user_to_topic ()

  This function subscribes the MQTT client to the topic specified allowing the user to recieve messages. This uses topic selection using a variable to decide on whether the topic the user should be listening to is their own topic or a groupchat topic. If they are in a groupchat, they will listen to the groupchat topic, if they are messaging one user, they will listen to their own topic.
- void check_recipient_type ()

  This function checks if the recipient is one user or a group of users, this then allows for correct topic selection. If the recipient is a group chat, the program will search for the users account permissions, to check if they are an admin or moderator of the group chat before loading the chat history. But if the recipient is just one user, the chat history will be loaded immediately.
- void read_chat_history ()

  This functions gets the users chat history between them and the recipient when the UI loads and displays it to the user. It requires two different SQL queries, the query is selected by checking the recipient type, this will allow the program to load the correct chat history from the database.
- void check_account_type ()

  This function checks if the current_account_type is set to Mod or Admin. If the current_account_type is set to Admin then the main menu button is hidden as the admin must not be able to message other users or join groups. However, if the current_account_type is set to Mod then the main_menu_button is displayed as Mods are able to message other users.
- void on_home_button_clicked ()

  This function hides the UI and shows the home page.
- void on_chats_button_clicked ()

  This function hides the UI and shows the chats page.
- void on_groups_button_clicked ()

  This function hides the UI and shows the groups page.
- void on_accounts_button_clicked ()

  This function hides the UI and shows the accounts page.
- void on_settings_button_clicked ()

This function hides the UI and shows the settings page.

- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.
- void on_manage_users_button_clicked ()

  This function hides the UI and shows the admin manage users page.
- void on_manage_groups_button_clicked ()

  This function hides the UI and shows the admin manage groups page.
- void on_manage_mods_button_clicked ()

  This function hides the UI and shows the admin manage mods page.
- void on_logoff_button_clicked ()

  This function is triggered once the user manually clicks the logoff_button or once the inactive_minutes global variable is = 15. An SQL query is run which sets the current users inactive status to 'Offline' in the database whilst redirecting them to the login page.
- void on_group_button_clicked ()

  This function hides the UI and shows the admin manage groups page, and sets a variable so it only allows group admin permissions.
- void on_write_message_cursorPositionChanged ()

  This function checks the input field for its cursor position, if it changes reset the inactivity settings.

Private Member Functions

- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.
- void keyPressEvent (QKeyEvent ∗event)

  This function checks for the user pressing keys, if so reset the inactivity settings.

Private Attributes

- Ui::MessagingPageNew ∗ **ui**
- QMqttClient ∗ **sdi_client**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

3.24.1 Member Function Documentation

3.24.1.1 on_send_button_clicked

```
void MessagingPageNew::on_send_button_clicked ( )  [private], [slot]
```

This function updates the users UI with the message they want to send and sends it to the recipient. For updating the users UI, the message is styled with a QTextBlockFormat to be left aligned and also styles the block to make message belonging more visible to the user. If the user is messaging one user the message date and time are sent, if they are messaging a groupchat, it sends the message, date and time and their username to show who sent the message in the groupchat.

The message is uploaded to the corressponding chat history table on the database, so the message history can be displayed to the user.

3.24.1.2 subscribe_user_to_topic

```
void MessagingPageNew::subscribe_user_to_topic ( )  [private], [slot]
```

This function subscribes the MQTT client to the topic specified allowing the user to recieve messages. This uses topic selection using a variable to decide on whether the topic the user should be listening to is their own topic or a groupchat topic. If they are in a groupchat, they will listen to the groupchat topic, if they are messaging one user, they will listen to their own topic.

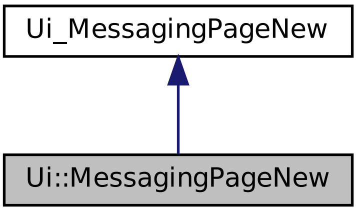
The subscribed topics are multi-level topics, these topics are filtered to only receive the correct topic in the chat everytime a message is received by the user.

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

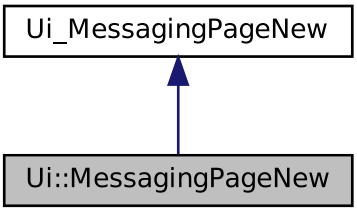
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/messagingpage.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/messagingpage.cpp

3.25 Ui::MessagingPageNew Class Reference

Inheritance diagram for Ui::MessagingPageNew:



Collaboration diagram for Ui::MessagingPageNew:



Additional Inherited Members

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_messagingpage.h

3.26 qt_meta_stringdata_Accounts_t Struct Reference

Public Attributes

- QByteArrayData **data** [20]
- char **stringdata0** [423]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_accounts.cpp

3.27 qt_meta_stringdata_AdminHome_t Struct Reference

Public Attributes

- QByteArrayData **data** [13]
- char **stringdata0** [295]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_adminhome.cpp

3.28 qt_meta_stringdata_AdminManageGroups_t Struct Reference

Public Attributes

- QByteArrayData **data** [23]
- char **stringdata0** [494]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_adminmanagegroups.cpp

3.29 qt_meta_stringdata_AdminManageMods_t Struct Reference

Public Attributes

- QByteArrayData **data** [18]
- char **stringdata0** [390]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_adminmanagemods.cpp

3.30 qt_meta_stringdata_AdminManageUsers_t Struct Reference

Public Attributes

- QByteArrayData **data** [18]
- char **stringdata0** [389]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_adminmanageusers.cpp

3.31 qt_meta_stringdata_AdminSettings_t Struct Reference

Public Attributes

- QByteArrayData **data** [8]
- char **stringdata0** [165]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_adminsettings.cpp

3.32 qt_meta_stringdata_ExistingChats_t Struct Reference

Public Attributes

- QByteArrayData **data** [20]
- char **stringdata0** [446]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_existingchats.cpp

3.33 qt_meta_stringdata_ForgottenPassword_t Struct Reference

Public Attributes

- QByteArrayData **data** [9]
- char **stringdata0** [162]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_forgottenpassword.cpp

3.34 qt_meta_stringdata_GroupChats_t Struct Reference

Public Attributes

- QByteArrayData **data** [28]
- char **stringdata0** [565]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_groupchats.cpp

3.35 qt_meta_stringdata_Groups_t Struct Reference

Public Attributes

- QByteArrayData **data** [21]
- char **stringdata0** [463]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_groups.cpp

3.36 qt_meta_stringdata_HomePageNewest_t Struct Reference

Public Attributes

- QByteArrayData **data** [16]
- char **stringdata0** [375]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_homepage.cpp

3.37 qt_meta_stringdata_MainWindow_t Struct Reference

Public Attributes

- QByteArrayData **data** [13]
- char **stringdata0** [233]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_mainwindow.cpp

3.38 qt_meta_stringdata_MessagingPageNew_t Struct Reference

Public Attributes

- QByteArrayData **data** [24]
- char **stringdata0** [571]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_messagingpage.cpp

3.39 qt_meta_stringdata_Register_t Struct Reference

Public Attributes

- QByteArrayData **data** [9]
- char **stringdata0** [139]

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_register.cpp

3.40 qt_meta_stringdata_Settings_t Struct Reference

Public Attributes

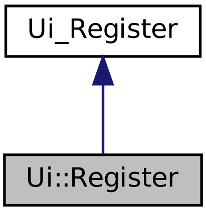
- QByteArrayData **data** [36]
- char **stringdata0** [834]

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

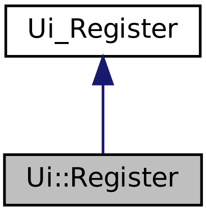
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/moc_settings.cpp

3.41 Ui::Register Class Reference

Inheritance diagram for Ui::Register:



Collaboration diagram for Ui::Register:



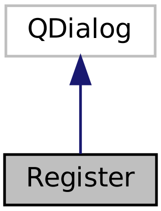
Additional Inherited Members

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

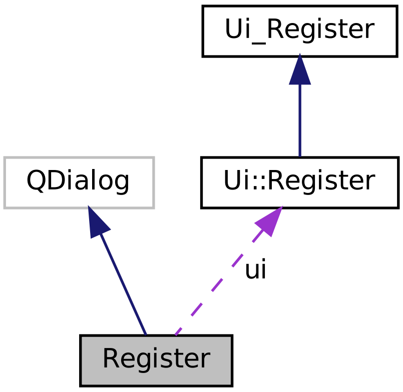
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_register.h

3.42 Register Class Reference

Inheritance diagram for Register:



Collaboration diagram for Register:



Public Member Functions

- **Register** (QWidget ∗parent=nullptr)

Private Slots

- void **update_date_time** ()
- void on_back_to_login_clicked ()

  login page is shown and the register page is hidden
- void on_register_button_clicked ()

Takes all of the user inputs into each lineedit, validates them by checking all details are filled in, makes sure the username is unique and password has at least 8 characters, special characters and a number. Once these have been confirmed they are saved to the database.

- void on_shutdown_button_clicked ()

  Closes the application.
- QString Encrypt (QString password_unencrypted, int key)

  Encrypts the user password using a Caesar cipher.

Private Member Functions

- void **mousePressEvent** (QMouseEvent ∗event)
- void **mouseMoveEvent** (QMouseEvent ∗event)

Private Attributes

- Ui::Register ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

3.42.1 Member Function Documentation

3.42.1.1 Encrypt

```
QString Register::Encrypt (
            QString password_unencrypted,
            int key )  [private], [slot]
```

Encrypts the user password using a Caesar cipher.

Parameters

password_unencrypted	
key	

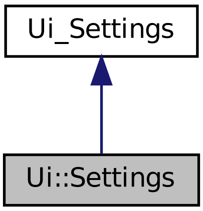
Returns

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

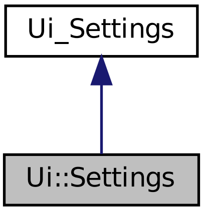
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/register.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/register.cpp

3.43 Ui::Settings Class Reference

Inheritance diagram for Ui::Settings:



Collaboration diagram for Ui::Settings:



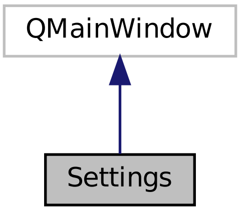
Additional Inherited Members

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

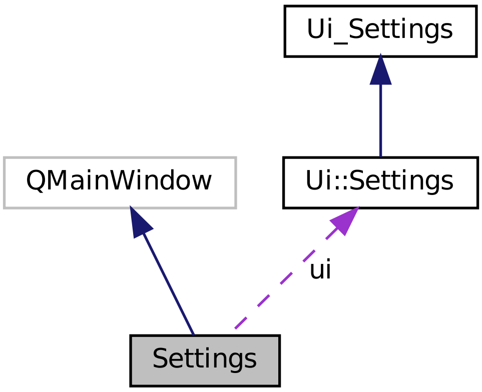
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_settings.h

3.44 Settings Class Reference

Inheritance diagram for Settings:



Collaboration diagram for Settings:



Public Member Functions

- **Settings** (QWidget ∗parent=nullptr)

Private Slots

- void update_inactivity_status ()

  This function is called every second that the timer adds one to it. Then, inactive_seconds get added by 1 and once they add up to 60, inactive_minutes gets set to 1. Lastly once inactive minutes add up to 15, then both inactive seconds and minutes get reset back to 0 while, the on_logoff_button_clicked function gets called and the user is automatically logged off and directed to the home page. This however, is prevented if at any point the user clicks anywhere on the page.

- void update_date_time ()

  This function updates the current date and time dynamically every second.
- void check_account_type ()

  This function checks if the current_account_type is set to Mod or Admin. If the current_account_type is set to Admin then the main menu button is hidden as the admin must not be able to message other users or join groups. However, if the current_account_type is set to Mod then the main_menu_button is displayed as Mods are able to message other users.
- void update_password ()

  This function updates the password with the password entered into the password input field if the password passes all the validation checks.
- void update_first_name ()

  This function first name the first name with the first name entered into the first name input field.
- void update_last_name ()

  This function last name the last name with the last name entered into the last name input field.
- void update_email ()

  This function updates the email with the email entered into the email input field if the email passes all the validation checks.
- QString Encrypt (QString password_unencrypted, int key)

  Encrypts the user password using a Caesar cipher.
- void update_special_word ()

  This function updates the special_word with the special_word entered into the special_word input field if the special↩ _word passes all the validation checks.
- void update_username ()

  This function updates the username with the username entered into the username input field if the username passes all the validation checks.
- void on_browse_button_clicked ()

  This function opens a file browser for you to select a jpg image file. After the file has been selected, the users profile picture is updated in the application and on the database.
- void on_home_button_clicked ()

  This function hides the UI and shows the home page.
- void on_chats_button_clicked ()

  This function hides the UI and shows the chats page.
- void on_groups_button_clicked ()

  This function hides the UI and shows the groups page.
- void on_accounts_button_clicked ()

  This function hides the UI and shows the accounts page.
- void on_settings_button_clicked ()

  This function hides the UI and shows the settings page.
- void on_shutdown_button_clicked ()

  This function is triggered once the user manually clicks the shutdown_button. An SQL query is run to set the current users active_status to 'Offline' as they have turned off the application.
- void on_manage_users_button_clicked ()

  This function hides the UI and shows the admin manage users page.
- void on_manage_groups_button_clicked ()

  This function hides the UI and shows the admin manage groups page.
- void on_manage_mods_button_clicked ()

  This function hides the UI and shows the admin manage mods page.
- void on_logoff_button_clicked ()

  This function is triggered once the user manually clicks the logoff_button or once the inactive_minutes global variable is = 15. An SQL query is run which sets the current users inactive status to 'Offline' in the database whilst redirecting them to the login page.
- void on_update_details_button_clicked ()

This function checks all the input fields, if they have data in them, it updates the database with the new information, if it passes all the validation needed for each field.

- void on_accessibility_button_clicked ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light, sets to the opposite setting and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.
- void on_new_username_cursorPositionChanged (int arg1, int arg2)

  This function checks if the input fields cursor position has changed, if so reset the inactivity settings.
- void on_new_password_cursorPositionChanged (int arg1, int arg2)

  This function checks if the input fields cursor position has changed, if so reset the inactivity settings.
- void on_new_first_name_cursorPositionChanged (int arg1, int arg2)

  This function checks if the input fields cursor position has changed, if so reset the inactivity settings.
- void on_new_last_name_cursorPositionChanged (int arg1, int arg2)

  This function checks if the input fields cursor position has changed, if so reset the inactivity settings.
- void on_new_email_cursorPositionChanged (int arg1, int arg2)

  This function checks if the input fields cursor position has changed, if so reset the inactivity settings.
- void on_new_special_word_cursorPositionChanged (int arg1, int arg2)

  This function checks if the input fields cursor position has changed, if so reset the inactivity settings.
- void set_accessibility_settings ()

  This function checks if the global function dark_mode_toggle is set to be Dark or light and then accordingly, displays the correct styling. Each UI element is targeted and has its stylesheet set to either grey colours for dark mode or white colours for the default light mode settings.

Private Member Functions

- void mousePressEvent (QMouseEvent ∗event)

  This function checks for the user pressing mouse buttons, if so reset the inactivity settings.
- void mouseMoveEvent (QMouseEvent ∗event)

  This function if the user has pressed their mouse button, they can move the window.
- void keyPressEvent (QKeyEvent ∗event)

  This function checks for the user pressing keys, if so reset the inactivity settings.

Private Attributes

- Ui::Settings ∗ **ui**
- QSqlDatabase **database**
- int **m_nMouseClick_X_Coordinate**
- int **m_nMouseClick_Y_Coordinate**

3.44.1 Member Function Documentation

3.44.1.1 Encrypt

```
QString Settings::Encrypt (
            QString password_unencrypted,
            int key )  [private], [slot]
```

Encrypts the user password using a Caesar cipher.

Parameters

password_unencrypted	
key	

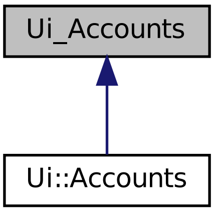
Returns

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/settings.h
- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/settings.cpp

3.45 Ui_Accounts Class Reference

Inheritance diagram for Ui_Accounts:



Public Member Functions

- void **setupUi** (QMainWindow ∗Accounts)
- void **retranslateUi** (QMainWindow ∗Accounts)

Public Attributes

- QWidget ∗ **central_widget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer_7**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name**

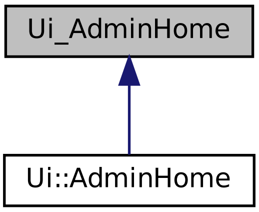
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button**
- QPushButton ∗ **shutdown_button**
- QHBoxLayout ∗ **Page**
- QWidget ∗ **sidebar**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button**
- QPushButton ∗ **chats_button**
- QPushButton ∗ **groups_button**
- QPushButton ∗ **accounts_button**
- QPushButton ∗ **settings_button**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**
- QPushButton ∗ **manage_users_button**
- QPushButton ∗ **manage_groups_button**
- QPushButton ∗ **manage_mods_button**
- QSpacerItem ∗ **verticalSpacer_6**
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **page_name_widget**
- QHBoxLayout ∗ **horizontalLayout_7**
- QLabel ∗ **page_title_label**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **page_contents_widget**
- QHBoxLayout ∗ **horizontalLayout_8**
- QSpacerItem ∗ **horizontalSpacer_2**
- QTableWidget ∗ **show_users_table**
- QSpacerItem ∗ **horizontalSpacer**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_accounts.h

3.46 Ui_AdminHome Class Reference

Inheritance diagram for Ui_AdminHome:



Public Member Functions

- void **setupUi** (QMainWindow ∗AdminHome)
- void **retranslateUi** (QMainWindow ∗AdminHome)

Public Attributes

- QWidget ∗ **central_widget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer_7**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name**
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button**
- QPushButton ∗ **shutdown_button**
- QHBoxLayout ∗ **Page**
- QWidget ∗ **sidebar**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button**
- QPushButton ∗ **manage_users_button**
- QPushButton ∗ **manage_groups_button**
- QPushButton ∗ **manage_mods_button**
- QPushButton ∗ **main_menu_button**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**
- QSpacerItem ∗ **verticalSpacer_6**

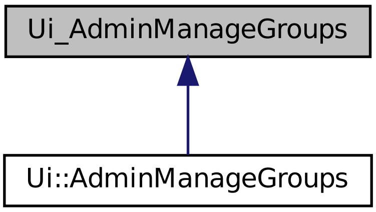
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **page_name_widget**
- QHBoxLayout ∗ **horizontalLayout_7**
- QLabel ∗ **page_title_label**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **page_contents_widget**
- QHBoxLayout ∗ **horizontalLayout_8**
- QTextBrowser ∗ **page_information**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_adminhome.h

3.47 Ui_AdminManageGroups Class Reference

Inheritance diagram for Ui_AdminManageGroups:



Public Member Functions

- void **setupUi** (QMainWindow ∗AdminManageGroups)
- void **retranslateUi** (QMainWindow ∗AdminManageGroups)

Public Attributes

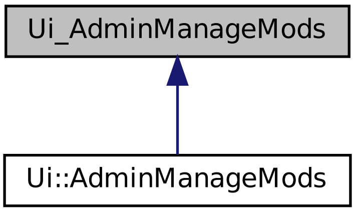
- QWidget ∗ **central_widget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **currentDateTime**
- QSpacerItem ∗ **horizontalSpacer_7**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name**
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button**
- QPushButton ∗ **shutdown_button**
- QHBoxLayout ∗ **Page**
- QWidget ∗ **sidebar**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button**
- QPushButton ∗ **manage_users_button**
- QPushButton ∗ **manage_groups_button**
- QPushButton ∗ **manage_mods_button**
- QPushButton ∗ **main_menu_button**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**
- QSpacerItem ∗ **verticalSpacer_6**
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **page_name_widget**
- QHBoxLayout ∗ **horizontalLayout_7**
- QLabel ∗ **page_title_label**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **page_contents_widget**
- QHBoxLayout ∗ **horizontalLayout_8**
- QVBoxLayout ∗ **verticalLayout_2**
- QTableWidget ∗ **tableWidget**
- QLabel ∗ **label**
- QSpacerItem ∗ **verticalSpacer_2**
- QWidget ∗ **button_widget**
- QVBoxLayout ∗ **verticalLayout_3**
- QPushButton ∗ **promote_button**
- QPushButton ∗ **demote_button**
- QPushButton ∗ **delete_button**
- QPushButton ∗ **remove_button**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_adminmanagegroups.h

3.48 Ui_AdminManageMods Class Reference

Inheritance diagram for Ui_AdminManageMods:



Public Member Functions

- void **setupUi** (QMainWindow ∗AdminManageMods)
- void **retranslateUi** (QMainWindow ∗AdminManageMods)

Public Attributes

- QWidget ∗ **central_widget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer_7**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name**
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button**
- QPushButton ∗ **shutdown_button**
- QHBoxLayout ∗ **Page**
- QWidget ∗ **sidebar**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button**
- QPushButton ∗ **manage_users_button**
- QPushButton ∗ **manage_groups_button**
- QPushButton ∗ **manage_mods_button**
- QPushButton ∗ **main_menu_button**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**
- QSpacerItem ∗ **verticalSpacer_6**

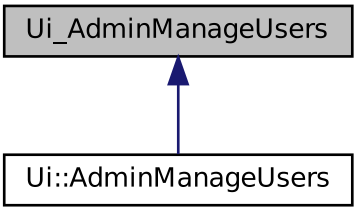
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **page_name_widget**
- QHBoxLayout ∗ **horizontalLayout_7**
- QLabel ∗ **page_title_label**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **page_contents_widget**
- QHBoxLayout ∗ **horizontalLayout_8**
- QVBoxLayout ∗ **verticalLayout_2**
- QTableWidget ∗ **tableWidget**
- QLabel ∗ **label**
- QSpacerItem ∗ **verticalSpacer_2**
- QWidget ∗ **button_widget**
- QVBoxLayout ∗ **verticalLayout_3**
- QPushButton ∗ **promote_button**
- QPushButton ∗ **demote_button**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_adminmanagemods.h

3.49 Ui_AdminManageUsers Class Reference

Inheritance diagram for Ui_AdminManageUsers:



Public Member Functions

- void **setupUi** (QMainWindow ∗AdminManageUsers)
- void **retranslateUi** (QMainWindow ∗AdminManageUsers)

Public Attributes

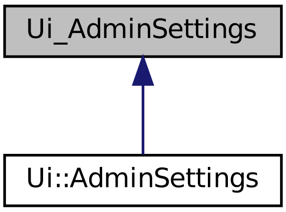
- QWidget ∗ **central_widget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer_7**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name**
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button**
- QPushButton ∗ **shutdown_button**
- QHBoxLayout ∗ **Page**
- QWidget ∗ **sidebar**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button**
- QPushButton ∗ **manage_users_button**
- QPushButton ∗ **manage_groups_button**
- QPushButton ∗ **manage_mods_button**
- QPushButton ∗ **main_menu_button**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**
- QSpacerItem ∗ **verticalSpacer_6**
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **page_name_widget**
- QHBoxLayout ∗ **horizontalLayout_7**
- QLabel ∗ **page_title_label**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **page_contents_widget**
- QHBoxLayout ∗ **horizontalLayout_8**
- QVBoxLayout ∗ **verticalLayout_2**
- QTableWidget ∗ **tableWidget**
- QLabel ∗ **label**
- QSpacerItem ∗ **verticalSpacer_2**
- QWidget ∗ **button_widget**
- QVBoxLayout ∗ **verticalLayout_3**
- QPushButton ∗ **delete_button**
- QPushButton ∗ **reset_button**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_adminmanageusers.h

3.50 Ui_AdminSettings Class Reference

Inheritance diagram for Ui_AdminSettings:



Public Member Functions

- void **setupUi** (QMainWindow ∗AdminSettings)
- void **retranslateUi** (QMainWindow ∗AdminSettings)

Public Attributes

- QWidget ∗ **centralwidget**
- QVBoxLayout ∗ **verticalLayout_2**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **currentDateTime_2**
- QSpacerItem ∗ **horizontalSpacer_7**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name_2**
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button_2**
- QPushButton ∗ **shutdown_button_2**
- QVBoxLayout ∗ **verticalLayout**
- QHBoxLayout ∗ **Page**
- QWidget ∗ **sidebar_2**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button_2**
- QPushButton ∗ **musers_button_2**
- QPushButton ∗ **mgroups_button_2**
- QPushButton ∗ **mmods_button_2**
- QPushButton ∗ **settings_button_2**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**

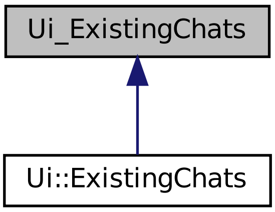
- QSpacerItem ∗ **verticalSpacer_6**
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **PageName_widget_2**
- QHBoxLayout ∗ **horizontalLayout_7**
- QLabel ∗ **page_title_label_2**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **Pagecontents_widget_2**
- QHBoxLayout ∗ **horizontalLayout_8**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_adminsettings.h

3.51 Ui_ExistingChats Class Reference

Inheritance diagram for Ui_ExistingChats:



Public Member Functions

- void **setupUi** (QMainWindow ∗ExistingChats)
- void **retranslateUi** (QMainWindow ∗ExistingChats)

Public Attributes

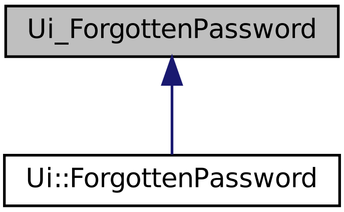
- QWidget ∗ **central_widget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer_7**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name**
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button**
- QPushButton ∗ **shutdown_button**
- QHBoxLayout ∗ **Page**
- QWidget ∗ **sidebar**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button**
- QPushButton ∗ **chats_button**
- QPushButton ∗ **groups_button**
- QPushButton ∗ **accounts_button**
- QPushButton ∗ **settings_button**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**
- QPushButton ∗ **manage_users_button**
- QPushButton ∗ **manage_groups_button**
- QPushButton ∗ **manage_mods_button**
- QSpacerItem ∗ **verticalSpacer_6**
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **page_name_widget**
- QHBoxLayout ∗ **horizontalLayout_7**
- QLabel ∗ **page_title_label**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **page_contents_widget**
- QHBoxLayout ∗ **horizontalLayout_8**
- QSpacerItem ∗ **horizontalSpacer**
- QTableWidget ∗ **show_existing_chats_table**
- QSpacerItem ∗ **horizontalSpacer_2**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_existingchats.h

3.52 Ui_ForgottenPassword Class Reference

Inheritance diagram for Ui_ForgottenPassword:



Public Member Functions

- void **setupUi** (QMainWindow ∗ForgottenPassword)
- void **retranslateUi** (QMainWindow ∗ForgottenPassword)

Public Attributes

- QWidget ∗ **centralwidget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **topPanel**
- QHBoxLayout ∗ **horizontalLayout**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer**
- QPushButton ∗ **shutdown_button**
- QSpacerItem ∗ **verticalSpacer_2**
- QHBoxLayout ∗ **horizontalLayout_2**
- QSpacerItem ∗ **horizontalSpacer_3**
- QWidget ∗ **forgot_password_form**
- QVBoxLayout ∗ **verticalLayout_2**
- QHBoxLayout ∗ **horizontalLayout_4**
- QLabel ∗ **username_label**
- QLineEdit ∗ **username**
- QHBoxLayout ∗ **horizontalLayout_6**
- QLabel ∗ **special_word_label**
- QLineEdit ∗ **special_word**
- QSpacerItem ∗ **verticalSpacer_3**
- QHBoxLayout ∗ **horizontalLayout_5**
- QLabel ∗ **new_password_label**
- QLineEdit ∗ **new_password**
- QHBoxLayout ∗ **horizontalLayout_3**
- QLabel ∗ **confirm_new_password_label**
- QLineEdit ∗ **confirm_new_password**
- QSpacerItem ∗ **verticalSpacer_4**
- QPushButton ∗ **change_password_button**

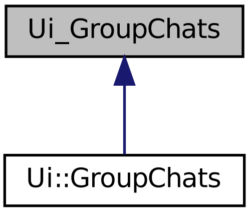
- QPushButton ∗ **back_to_login_button**
- QSpacerItem ∗ **horizontalSpacer_2**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_forgottenpassword.h

3.53 Ui_GroupChats Class Reference

Inheritance diagram for Ui_GroupChats:



Public Member Functions

- void **setupUi** (QMainWindow ∗GroupChats)
- void **retranslateUi** (QMainWindow ∗GroupChats)

Public Attributes

- QWidget ∗ **central_widget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer_7**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name**
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button**
- QPushButton ∗ **shutdown_button**
- QHBoxLayout ∗ **Page**

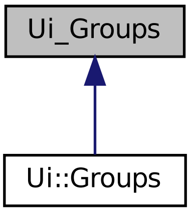
- QWidget ∗ **sidebar**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button**
- QPushButton ∗ **chats_button**
- QPushButton ∗ **groups_button**
- QPushButton ∗ **accounts_button**
- QPushButton ∗ **settings_button**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**
- QPushButton ∗ **manage_users_button**
- QPushButton ∗ **manage_groups_button**
- QPushButton ∗ **manage_mods_button**
- QSpacerItem ∗ **verticalSpacer_6**
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **page_name_widget**
- QHBoxLayout ∗ **horizontalLayout_7**
- QLabel ∗ **page_title_label**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **page_contents_widget**
- QHBoxLayout ∗ **horizontalLayout_8**
- QSpacerItem ∗ **horizontalSpacer**
- QHBoxLayout ∗ **horizontalLayout**
- QVBoxLayout ∗ **verticalLayout_2**
- QLineEdit ∗ **set_group_name**
- QTableWidget ∗ **add_remove_user_table**
- QSpacerItem ∗ **verticalSpacer_4**
- QWidget ∗ **create_group_widget**
- QHBoxLayout ∗ **horizontalLayout_2**
- QPushButton ∗ **create_group_button**
- QSpacerItem ∗ **verticalSpacer_3**
- QSpacerItem ∗ **horizontalSpacer_2**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_groupchats.h

3.54 Ui_Groups Class Reference

Inheritance diagram for Ui_Groups:



Public Member Functions

- void **setupUi** (QMainWindow ∗Groups)
- void **retranslateUi** (QMainWindow ∗Groups)

Public Attributes

- QWidget ∗ **central_widget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer_7**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name**
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button**
- QPushButton ∗ **shutdown_button**
- QHBoxLayout ∗ **Page**
- QWidget ∗ **sidebar**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button**
- QPushButton ∗ **chats_button**
- QPushButton ∗ **groups_button**
- QPushButton ∗ **accounts_button**
- QPushButton ∗ **settings_button**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**
- QPushButton ∗ **manage_users_button**

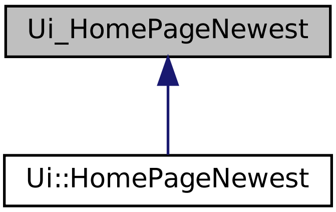
- QPushButton ∗ **manage_groups_button**
- QPushButton ∗ **manage_mods_button**
- QSpacerItem ∗ **verticalSpacer_6**
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **page_name_widget**
- QHBoxLayout ∗ **horizontalLayout_7**
- QSpacerItem ∗ **horizontalSpacer_4**
- QLabel ∗ **page_title_label**
- QSpacerItem ∗ **horizontalSpacer_3**
- QWidget ∗ **create_groups_widget**
- QHBoxLayout ∗ **horizontalLayout**
- QPushButton ∗ **create_group_button**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **page_contents_widget**
- QHBoxLayout ∗ **horizontalLayout_8**
- QSpacerItem ∗ **horizontalSpacer**
- QTableWidget ∗ **show_groupchats_table**
- QSpacerItem ∗ **horizontalSpacer_2**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_groups.h

3.55 Ui_HomePageNewest Class Reference

Inheritance diagram for Ui_HomePageNewest:



Public Member Functions

- void **setupUi** (QMainWindow ∗HomePageNewest)
- void **retranslateUi** (QMainWindow ∗HomePageNewest)

Public Attributes

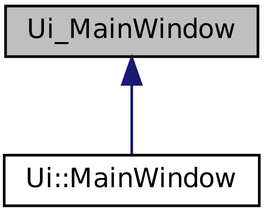
- QWidget ∗ **central_widget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer_7**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name**
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button**
- QPushButton ∗ **shutdown_button**
- QHBoxLayout ∗ **Page**
- QWidget ∗ **sidebar**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button**
- QPushButton ∗ **chats_button**
- QPushButton ∗ **groups_button**
- QPushButton ∗ **accounts_button**
- QPushButton ∗ **settings_button**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**
- QPushButton ∗ **manage_users_button**
- QPushButton ∗ **manage_groups_button**
- QPushButton ∗ **manage_mods_button**
- QSpacerItem ∗ **verticalSpacer_6**
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **page_name_widget**
- QHBoxLayout ∗ **horizontalLayout_7**
- QLabel ∗ **page_title_label**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **page_contents_widget**
- QHBoxLayout ∗ **horizontalLayout_8**
- QTextBrowser ∗ **page_information**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_homepage.h

3.56 Ui_MainWindow Class Reference

Inheritance diagram for Ui_MainWindow:



Public Member Functions

- void **setupUi** (QMainWindow ∗MainWindow)
- void **retranslateUi** (QMainWindow ∗MainWindow)

Public Attributes

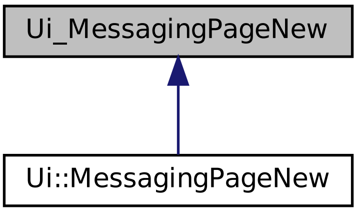
- QWidget ∗ **centralwidget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **topPanel**
- QHBoxLayout ∗ **horizontalLayout**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer**
- QPushButton ∗ **shutdown_button**
- QSpacerItem ∗ **verticalSpacer_3**
- QHBoxLayout ∗ **horizontalLayout_3**
- QWidget ∗ **loginForm**
- QVBoxLayout ∗ **verticalLayout_2**
- QHBoxLayout ∗ **horizontalLayout_5**
- QLabel ∗ **sername_label**
- QLineEdit ∗ **username**
- QHBoxLayout ∗ **horizontalLayout_4**
- QLabel ∗ **password_label**
- QLineEdit ∗ **password**
- QSpacerItem ∗ **verticalSpacer_2**
- QPushButton ∗ **login_button**
- QPushButton ∗ **register_button**
- QPushButton ∗ **forgotten_password_button**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_mainwindow.h

3.57 Ui_MessagingPageNew Class Reference

Inheritance diagram for Ui_MessagingPageNew:



Public Member Functions

- void **setupUi** (QMainWindow ∗MessagingPageNew)
- void **retranslateUi** (QMainWindow ∗MessagingPageNew)

Public Attributes

- QWidget ∗ **central_widget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer_7**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name**
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button**
- QPushButton ∗ **shutdown_button**
- QHBoxLayout ∗ **Page**
- QWidget ∗ **sidebar**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button**
- QPushButton ∗ **chats_button**
- QPushButton ∗ **groups_button**
- QPushButton ∗ **accounts_button**
- QPushButton ∗ **settings_button**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**
- QPushButton ∗ **manage_users_button**

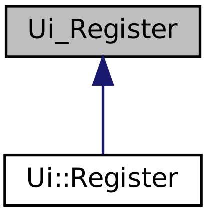
- QPushButton ∗ **manage_groups_button**
- QPushButton ∗ **manage_mods_button**
- QSpacerItem ∗ **verticalSpacer_6**
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **page_contents_widget**
- QHBoxLayout ∗ **horizontalLayout_5**
- QLabel ∗ **recipient_name**
- QSpacerItem ∗ **horizontalSpacer_3**
- QLabel ∗ **active_status**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **page_contents_widget_2**
- QHBoxLayout ∗ **horizontalLayout_8**
- QGridLayout ∗ **gridLayout**
- QPushButton ∗ **group_button**
- QSpacerItem ∗ **verticalSpacer_2**
- QTextBrowser ∗ **display_messages**
- QSpacerItem ∗ **verticalSpacer_3**
- QSpacerItem ∗ **horizontalSpacer_2**
- QTextEdit ∗ **write_message**
- QPushButton ∗ **send_button**
- QSpacerItem ∗ **horizontalSpacer**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_messagingpage.h

3.58 Ui_Register Class Reference

Inheritance diagram for Ui_Register:



Public Member Functions

- void **setupUi** (QDialog ∗Register)
- void **retranslateUi** (QDialog ∗Register)

Public Attributes

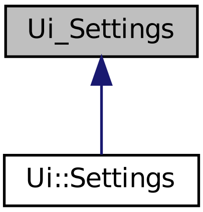
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **topPanel**
- QHBoxLayout ∗ **horizontalLayout_4**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer**
- QPushButton ∗ **shutdown_button**
- QSpacerItem ∗ **verticalSpacer_2**
- QHBoxLayout ∗ **horizontalLayout_5**
- QWidget ∗ **RegisterForm**
- QVBoxLayout ∗ **verticalLayout_2**
- QHBoxLayout ∗ **horizontalLayout_6**
- QLabel ∗ **Username_label**
- QLineEdit ∗ **username_input**
- QHBoxLayout ∗ **horizontalLayout_7**
- QLabel ∗ **Password_label**
- QLineEdit ∗ **password_input**
- QHBoxLayout ∗ **horizontalLayout_8**
- QLabel ∗ **Confirm_password_label**
- QLineEdit ∗ **confirm_password_input**
- QSpacerItem ∗ **verticalSpacer_3**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **FirstName_label**
- QLineEdit ∗ **first_name_input**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **LastName_label**
- QLineEdit ∗ **last_name_input**
- QHBoxLayout ∗ **horizontalLayout_11**
- QLabel ∗ **Email_label**
- QLineEdit ∗ **email_input**
- QHBoxLayout ∗ **horizontalLayout**
- QLabel ∗ **Special_Word_label**
- QLineEdit ∗ **special_word_input**
- QSpacerItem ∗ **verticalSpacer_4**
- QPushButton ∗ **register_button**
- QPushButton ∗ **back_to_login**
- QSpacerItem ∗ **verticalSpacer**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_register.h

3.59 Ui_Settings Class Reference

Inheritance diagram for Ui_Settings:



Public Member Functions

- void **setupUi** (QMainWindow ∗Settings)
- void **retranslateUi** (QMainWindow ∗Settings)

Public Attributes

- QWidget ∗ **central_widget**
- QVBoxLayout ∗ **verticalLayout**
- QWidget ∗ **TopWidget**
- QHBoxLayout ∗ **horizontalLayout_9**
- QLabel ∗ **current_date_time**
- QSpacerItem ∗ **horizontalSpacer_7**
- QPushButton ∗ **accessibility_button**
- QWidget ∗ **account_toppanel_button_2**
- QHBoxLayout ∗ **horizontalLayout_10**
- QLabel ∗ **account_name**
- QLabel ∗ **profile_pic**
- QSpacerItem ∗ **horizontalSpacer_8**
- QPushButton ∗ **logoff_button**
- QPushButton ∗ **shutdown_button**
- QHBoxLayout ∗ **Page**
- QWidget ∗ **sidebar**
- QVBoxLayout ∗ **verticalLayout_6**
- QWidget ∗ **sidemenu_topbar_2**
- QVBoxLayout ∗ **verticalLayout_7**
- QPushButton ∗ **home_button**
- QPushButton ∗ **chats_button**
- QPushButton ∗ **groups_button**
- QPushButton ∗ **accounts_button**
- QPushButton ∗ **settings_button**
- QSpacerItem ∗ **verticalSpacer_5**
- QWidget ∗ **sidemen_bottom_2**
- QVBoxLayout ∗ **verticalLayout_8**

- QPushButton ∗ **manage_users_button**
- QPushButton ∗ **manage_groups_button**
- QPushButton ∗ **manage_mods_button**
- QSpacerItem ∗ **verticalSpacer_6**
- QSpacerItem ∗ **horizontalSpacer_5**
- QVBoxLayout ∗ **InnerPage**
- QWidget ∗ **page_name_widget**
- QHBoxLayout ∗ **horizontalLayout_7**
- QLabel ∗ **page_title_label**
- QSpacerItem ∗ **verticalSpacer_7**
- QWidget ∗ **page_contents_widget**
- QHBoxLayout ∗ **horizontalLayout_8**
- QHBoxLayout ∗ **horizontalLayout**
- QSpacerItem ∗ **horizontalSpacer_10**
- QWidget ∗ **setting_left_widget**
- QVBoxLayout ∗ **verticalLayout_3**
- QSpacerItem ∗ **verticalSpacer_10**
- QHBoxLayout ∗ **horizontalLayout_13**
- QLabel ∗ **username_label**
- QLineEdit ∗ **new_username**
- QHBoxLayout ∗ **horizontalLayout_2**
- QLabel ∗ **password_label**
- QLineEdit ∗ **new_password**
- QHBoxLayout ∗ **horizontalLayout_3**
- QLabel ∗ **label_2**
- QLineEdit ∗ **new_first_name**
- QHBoxLayout ∗ **horizontalLayout_4**
- QLabel ∗ **label_3**
- QLineEdit ∗ **new_last_name**
- QHBoxLayout ∗ **horizontalLayout_5**
- QLabel ∗ **label_4**
- QLineEdit ∗ **new_email**
- QHBoxLayout ∗ **horizontalLayout_6**
- QLabel ∗ **special_word_label**
- QLineEdit ∗ **new_special_word**
- QHBoxLayout ∗ **horizontalLayout_11**
- QSpacerItem ∗ **horizontalSpacer**
- QPushButton ∗ **update_details_button**
- QSpacerItem ∗ **horizontalSpacer_2**
- QHBoxLayout ∗ **horizontalLayout_12**
- QSpacerItem ∗ **horizontalSpacer_4**
- QPushButton ∗ **browse_button**
- QSpacerItem ∗ **horizontalSpacer_9**
- QSpacerItem ∗ **horizontalSpacer_11**
- QSpacerItem ∗ **horizontalSpacer_6**
- QSpacerItem ∗ **verticalSpacer**
- QMenuBar ∗ **menubar**
- QStatusBar ∗ **statusbar**

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

- /home/ntu-user/eclipse-workspace/group_16_coursework/SDIApp/ui_settings.h

Index