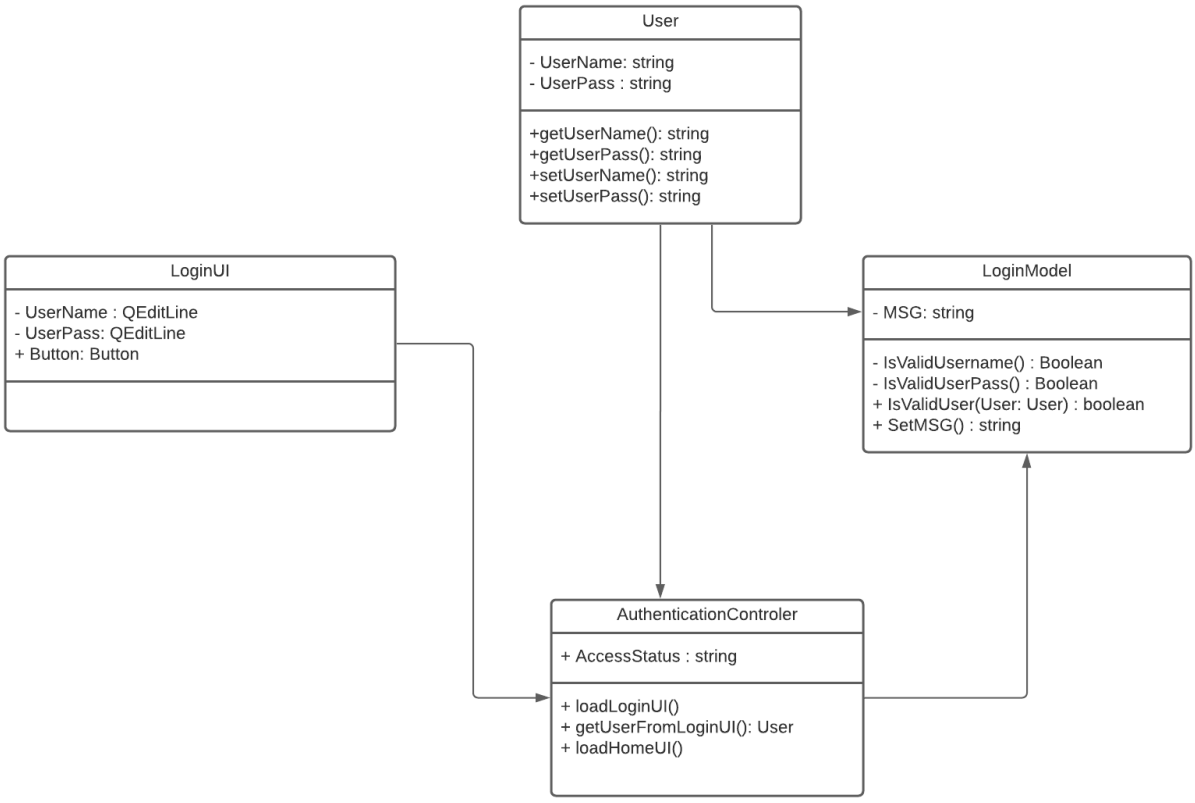
MARCH 16, 2021

SAFE HOME PROJECT

ZEXMAL AROJX

Zeemal Urooj
Ali Raza khan
Zia ur rehman
Sidra Ayesha
Rehan ali

LOGIN Class:



Description:

The description of the MVC is written below:

1. Model:

Model part consist of all the logical operation which a system performs when a user wants to perform any task. Task which user want to operate depends on various small functions and that small functions are written in the model part. Functions which are part of my model class are:

- IsValidUserName()
- IsValidUserPass() etc

2. View:

View is the variable part which we use to store information. All the information which we need from the user is fetched in these variables and then this information is used in various operations to perform particular task. Some of the variables which we use are:

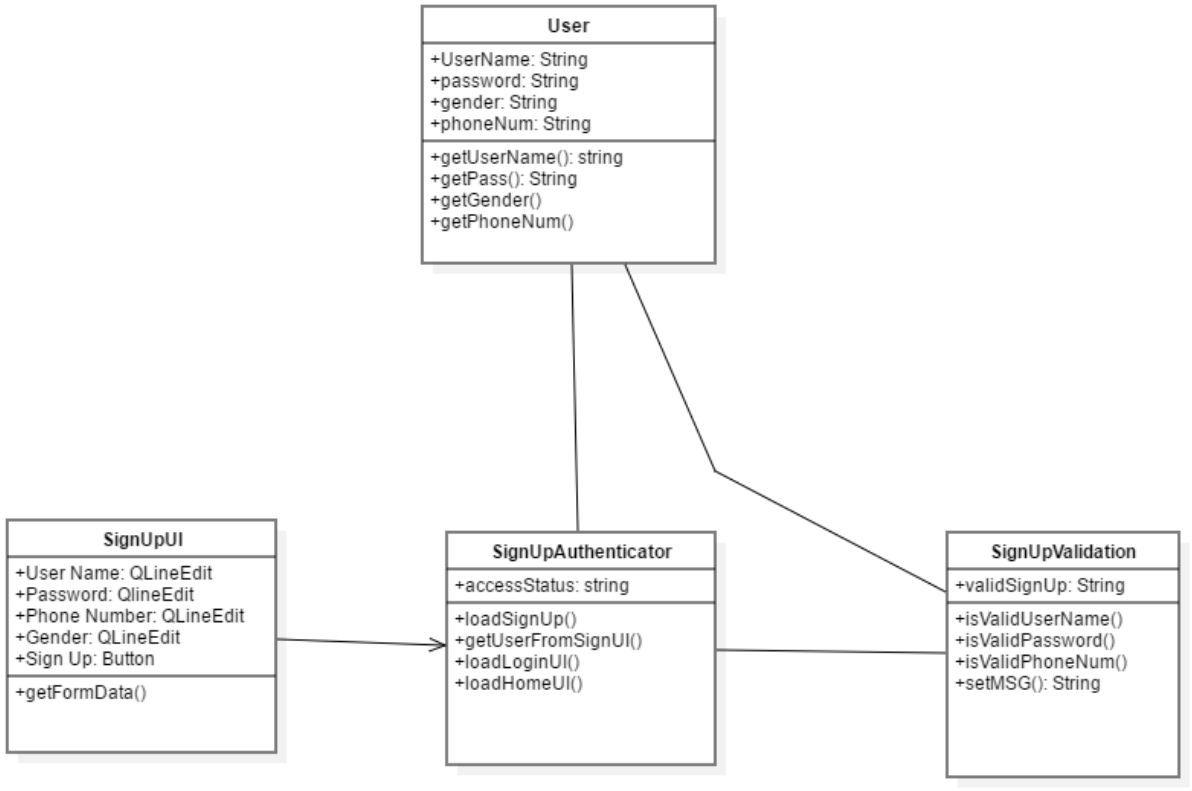
- UserPass
- UserName etcs

3. Controller:

Controller is the intermediate part between both of view and model. Model does not access to the variables of view directly, it has to take information from the controller part and control fetch the information from view class. Some of the functions of our controller part are:

- loadLoginUi
- getUserFromLoginUI() etc

Signup Class:



Description:

The description of the MVC is written below:

4. Model:

Model part consist of all the logical operation which a system performs when a user wants to perform any task. Task which user want to operate depends on various small functions and that small functions are written in the model part. Functions which are part of my model class for sign up are:

- IsValidphoneNumber()
- setMSG() etc

5. View:

View is the variable part which we use to store information. All the information which we need from the user is fetched in these variables and then this information is used in various operations to perform particular task. Some of the variables which we use are:

- Phone Number
- Gender etcs

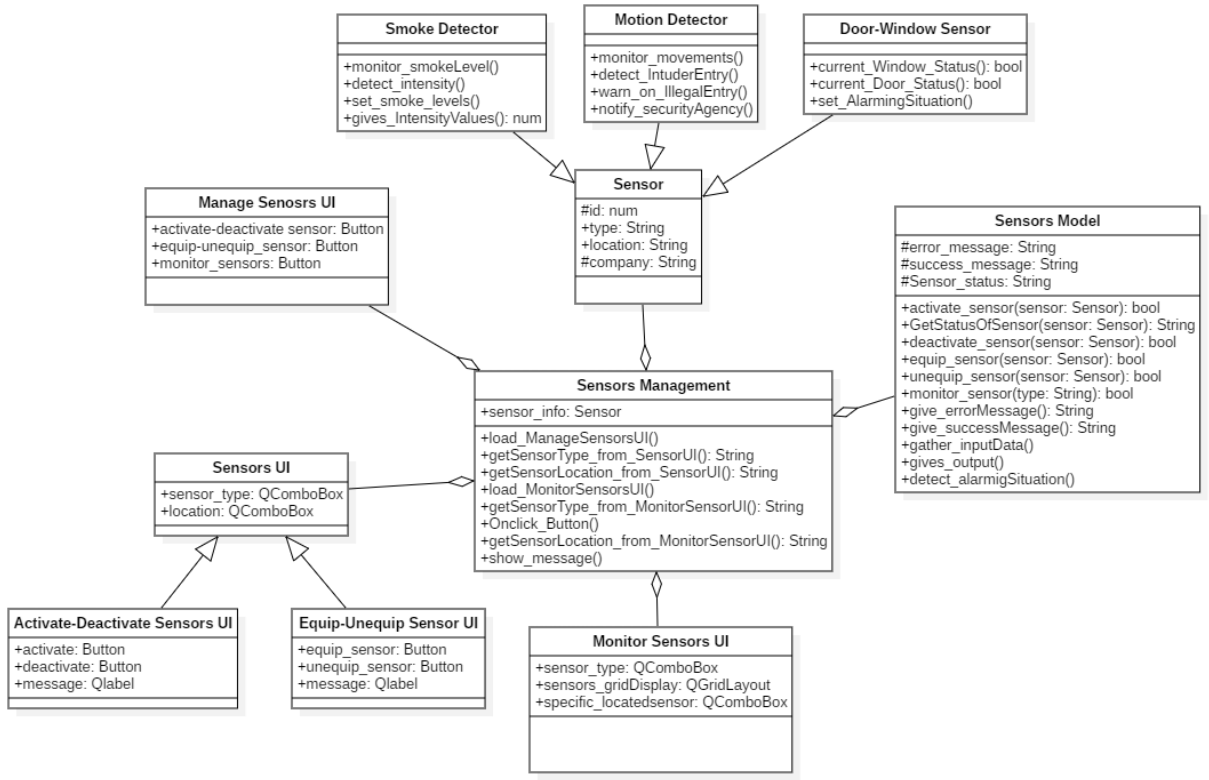
6. Controller:

Controller is the intermediate part between both of view and model. Model does not access to the variables of view directly, it has to take information from the controller part and control fetch the information from view class. Some of the functions of our controller part are:

- LoadSignUp()
- loadHome() etc

System features:

Sensor class:



Manage Sensor UI:

This is basically model class used as an interface. On this interface there will be menu of three buttons regarding sensor management. User can move on to specific action for example for activate sensor he may click on “Activate/Deactivate” button.

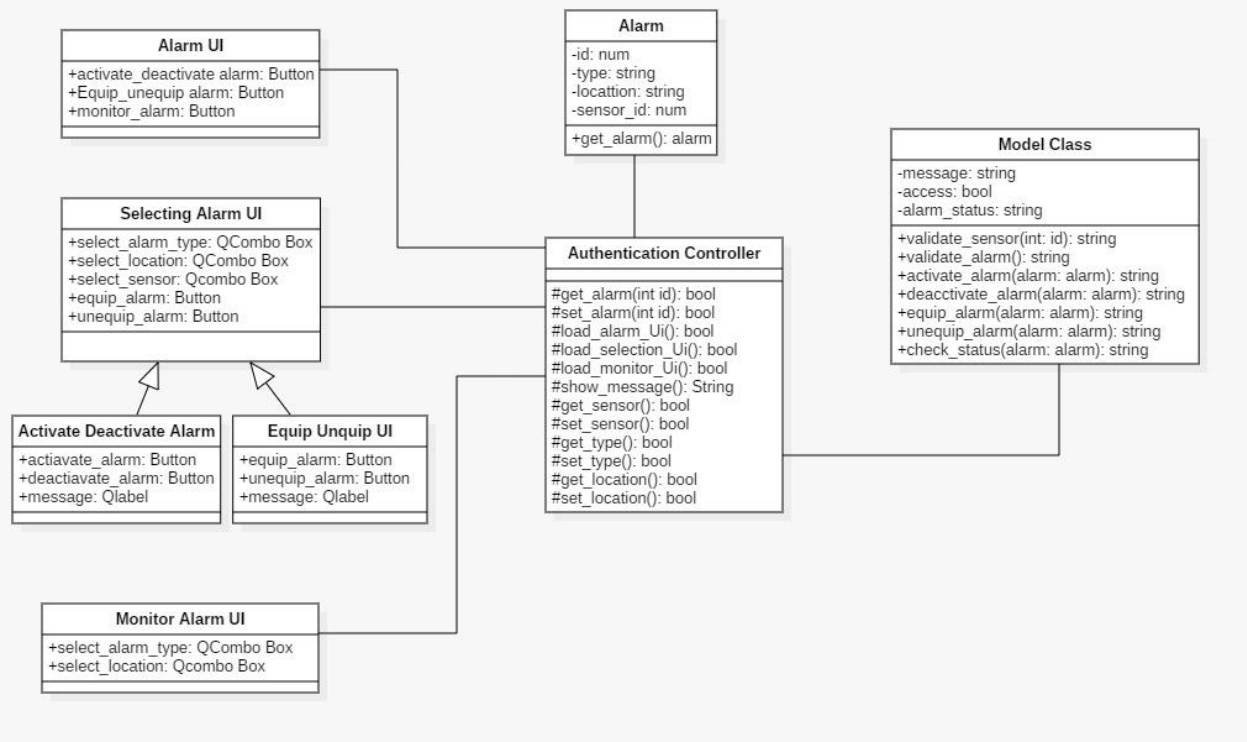
Sensors UI:

In this class we have to select sensor type from Combo box that select on which type of sensors we want to perform operation and the select location also from combo box these inputs are sent to controller class.

Activate/ Deactivate Sensors UI:

This class is inherited from Sensors UI class in which UI updated along with activate/ deactivate buttons and message will display after performing action successful message or error as message.

- ***Alarm class:***



ALARM UI

Alarm ui is main main interface model class. its interface of Alarm which basically handles the front end of this alarm system. There are several options like activate/deactivate, equip/unequip, and monitoring alarm so any one of them can be managed by specified buttons. These buttons are entry handle through which user can carry out the operation.

SELECTING ALARM UI

Selecting alarm UI have also many options or combo boxes that defines which type of operations have to perform regarding alarm, involving setting alarm location, alarm type, selecting sensors, equip alarm too.

EQUIP / UN-EQUIP

Equip and un equip action can also be performed through buttons. So this class is basically inherited by main “selecting Alarm UI”. So this is controlled by main class.

Activate/ Deactivate Alarm

Activation and Deactivation of alarm are also inherited by selecting Alarm UI. A status signal is also carried out after any activation or deactivation process of alarm.

ALARM USER INTERFACE

There is an option for user whether he wants to set a specific type of location of alarm so from many choices user can set by own choice so this is monitoring alarm user interface.

ALARM

This is main ALARM class. with different type of attributes as well, like alarm id, type , sensor id and alarm location is managed here. So this class has access through all type of functions and types regarding alarm system.

AUTHENTICATION CONTROLLER

This is main controller class of alarm. It has data about all functions, attributes and their associations as well.

It has many operations controls like get Alarm all data, setting the alarm properly by setting specific type and attributes.

It perform operation of loading alarm and its Ui, represents its complete view.

It also loades monitor Ui through which user can check the all working of alarm system.

It also gets and sets the type of alarm and its location too.

A signal is also generated after any specif little action of alarm.

Model Class

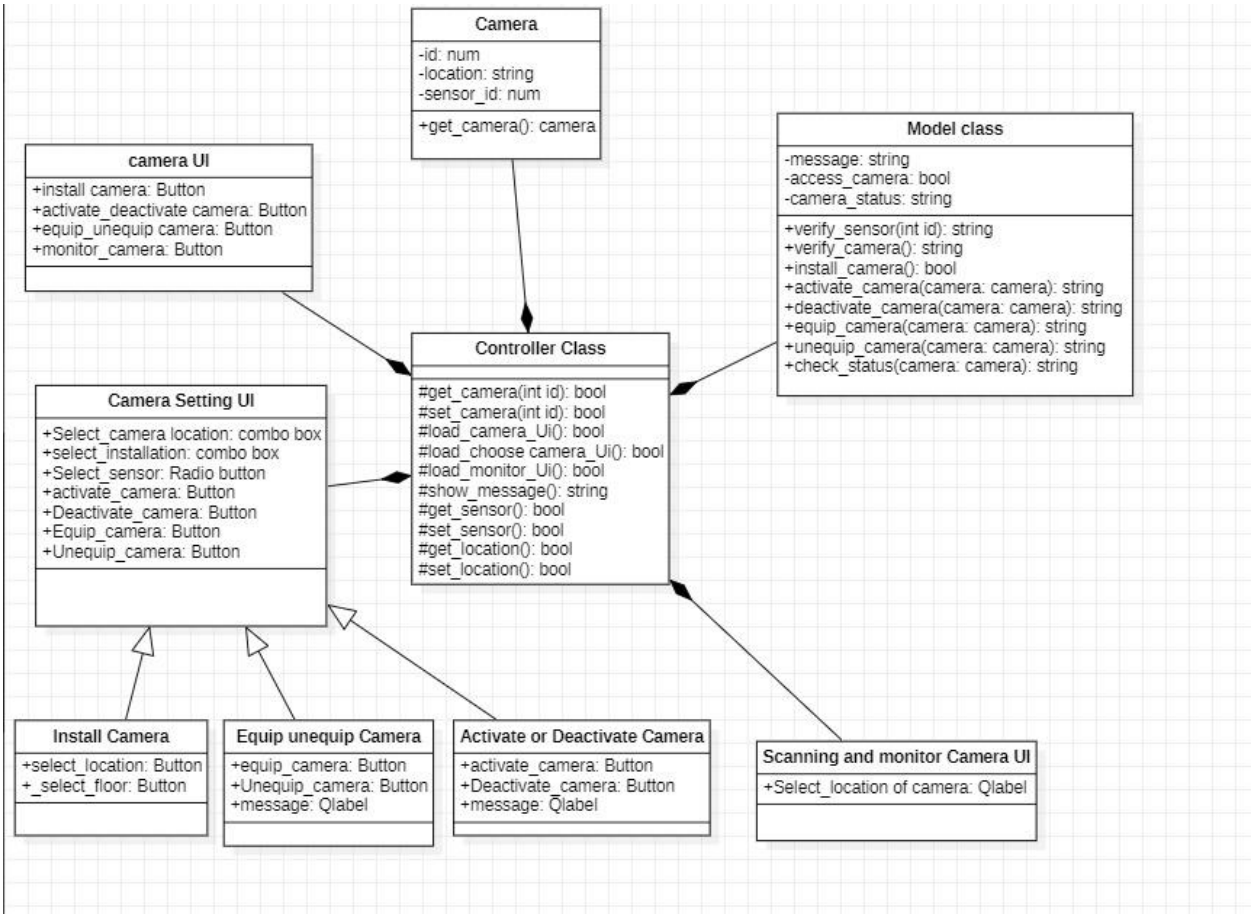
As ui controls the front end of any system so model class also controls the all backend process of alarm or any specific system.

Model class is very important class component and has an complete access to functions as well.

Like validation of sensors, setting types, equip/un-equip, activate/de-activate alarm, checking alarm status all is performed by model class.

Model class is combined to control class of alarm and its requirements about functions are also done with the help and association of control class.

- ***Camera Class:***



EXPLANATION OF (CAMERA) COMPONENT DIAGRAM

About Components:

In Safe Home project there are some main components, authentication, profile management, security and surveillance.

Camera View:

Camera is basic component of daily life too. In security factors, its important need to shield and monitor any specific system.

Camera is class of surveillance component. Camera is main component so it is managed by a profile with specific attributes, types and access function. Camera class is managed by its specific id, location and sensor id, and access to camera operations of system.

Camera UI:

Camera user interface has some specific functions like installation of camera, activate, deactivate and monitor.

Camera UI class is basically use as an interface so interface has many options and choices to jump any option. So camera UI has access to perform camera installation, activate or deactivate camera, equip or UN-equip camera through different choices performed by specific buttons.

Camera setting user interface has also different operations, including, selecting camera location, installation, selecting sensor, and activation/deactivation and equip/un-equip camera. Installation camera class, equip/un-equip, activate/de-activate are classes of camera setting UI, and inherited by it.

Basically setting camera ui provides an option or choices to select which location of camera has to be set.

Select installation of camera through combo box. Selecting and activation and deactivation all are managed by different choices so user can set any choice through different radio or combo buttons.

- Install Camera:

Install camera class is basically inherited by camera setting user interface. In install camera class location and floor has to be set by different buttons as well.

- Equip/UN equip Camera

Equip camera, UN equip camera, and after any updating or action, message or signal generation is one the main operations of the equip/unequip camera class. It is inherited by camera setting UI.

- Activate/ Deactivate Camera

This class is also inherited and deactivation or activation can be performed by it. And after any changing or any new updation, message passing has also an option of this class.

Scanning and Monitor Camera UI

This class is combined by controller class. And it has to be perform scanning camera component system and monitor it properly through different label.

Model Class:

Model class have different attributes like show message about camera, its access and status of camera. Model class have methods of verification of complete modules like verification, access, status, and activation/de M activation too. Basically Model class controls the all Backend working and UI controls the front end working of any component of major system.

Model class has some functions which help to control and manage the whole system model.

Controller Class:

Controller class is dominant and has access to all classes and controls them properly. Model class, Camera UI, setting Camera UI, Scanning and monitor Camera UI all are composed by Control class.

Controller class is like bridge between all classes and their associated components, any class which require specific operation control or specific action, communicates first with controller class.

Main camera class is also associated by Control class. So they all are linked eventually.

Component diagram: