

---

# Component Level Design

(Software Engineering)

**FALL 2020**



**NAMAL INSTITUTE**

Submitted to: Sir Jaleel  
Submitted by: SE-Project-SUNYC  
Dated: March 15th, 2021.



---

## Group Members:

No.	Name	Roll Number	Role
1	Mahmood Yousaf	1802011	Supervisor
2	Shahrukh Khan	1802029	Team-1
3	Wareesha Habib	1802006	Team-1
4	Usama Ahmad	1802027	Team-2
5	Noor Nabi	1802002	Team-2

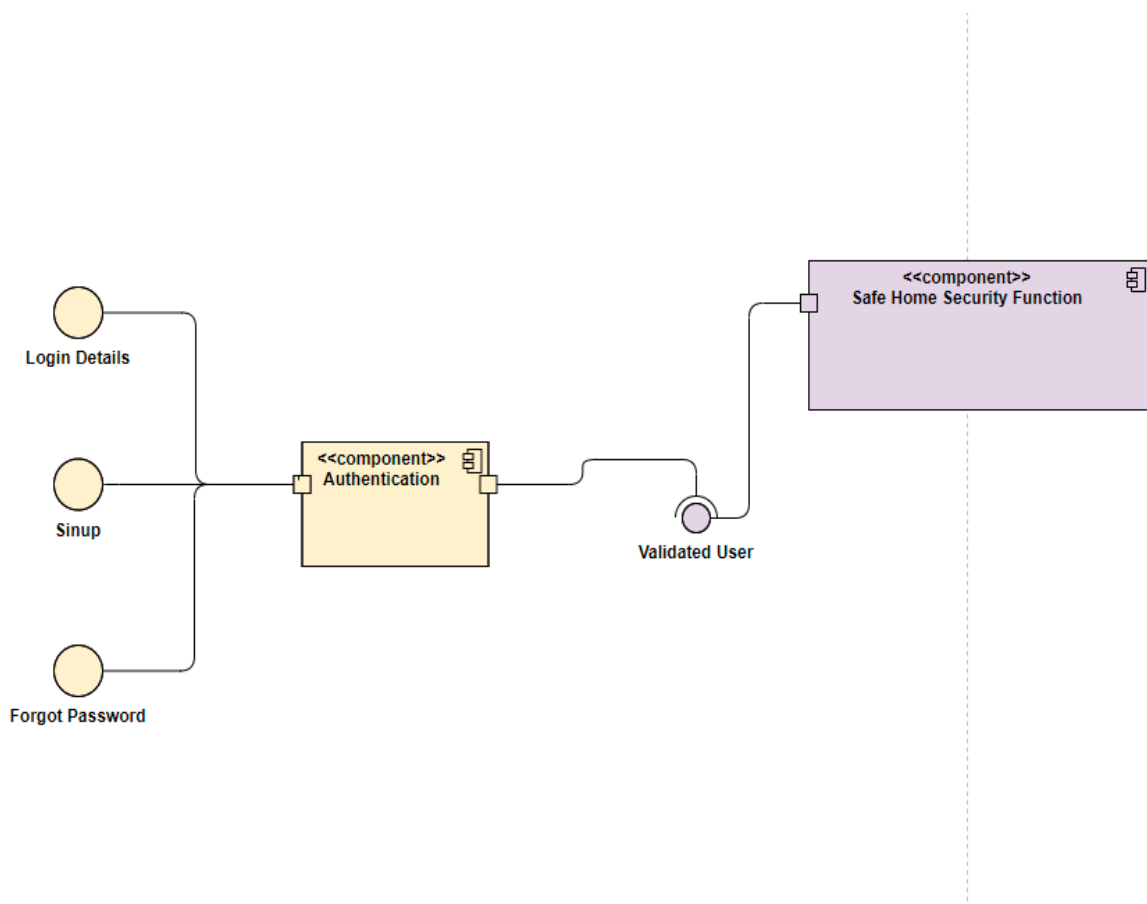


## Component Level Design

Component level design is the definition and design of components and modules after the architectural design phase. Component-level design defines the data structures, algorithms, interface characteristics, and communication mechanisms allocated to each component for the system development.

A complete set of software components is defined during architectural design. But the internal data structures and processing details of each component are not represented at a level of abstraction that is close to code. Component-level design defines the data structures, algorithms, interface characteristics, and communication mechanisms allocated to each component.

**According to OMG UML specification component is expressed as, “A modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces.”**



*Figure 1: Safe Home Component Level Design*



## Component Level Design Description

- Through interfaces input is passed to authentication component
- Authentication component validate the user by using user id and password
- After passing through authentication component, user is validated for using safe home security function
- Without passing to authentication component user will not able to use safe home security function





## Authentication Component by Team-2

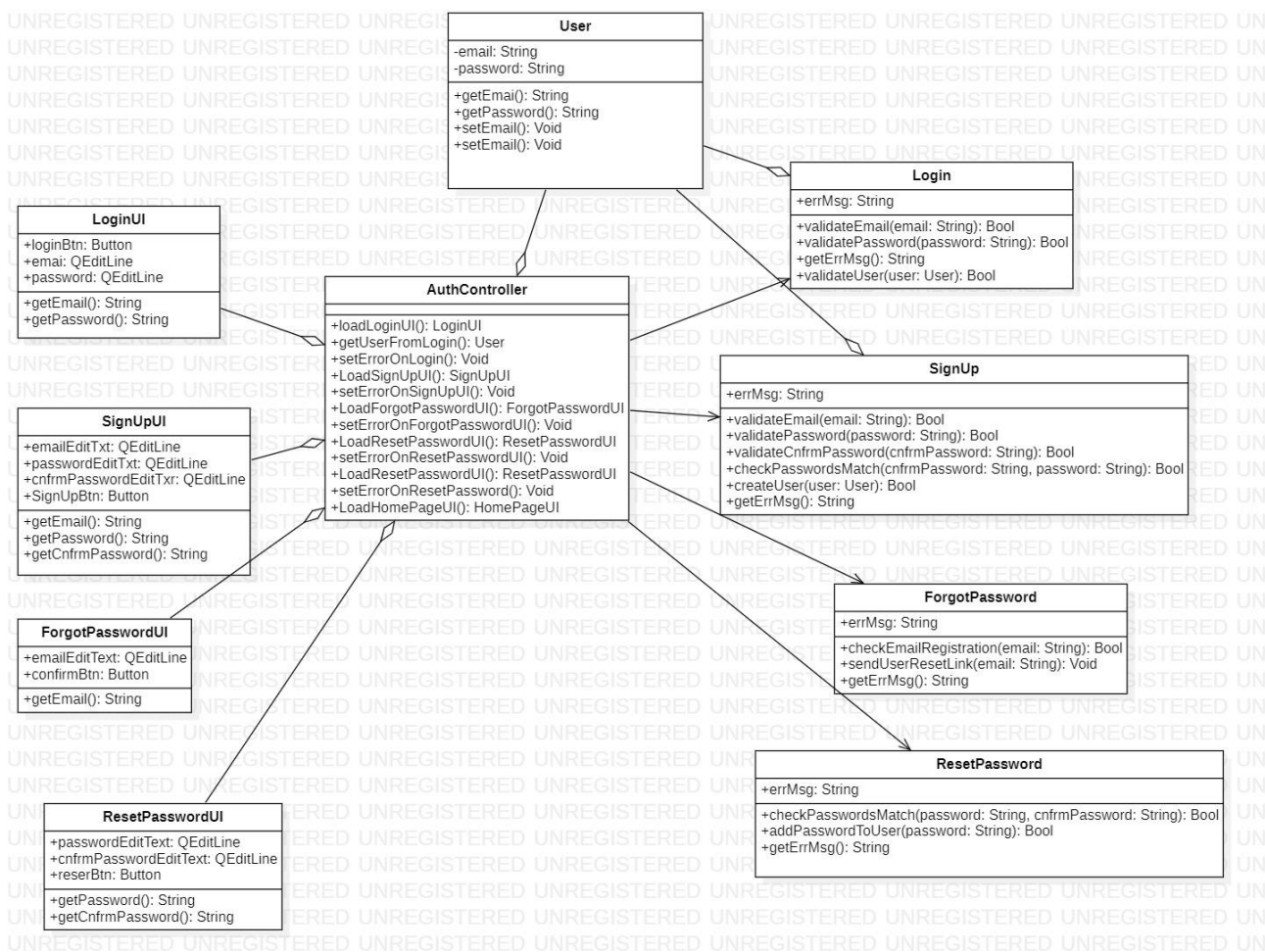


Figure 2: Authentication Component



## Description Authentication Component

- There are four UI classes, Sign In, SignUp, ForgotPassword, ResetPassword all the classes have only 2 UI components QEditLine and Button. In SignInUI class it has two Fields of input which one for email and other for the password and a button to submit the form. There are getters which are used to get the email and password from the respective fields.
- The second UI class is of SignUpUI it has 3 input fields one for email one for password and one for reentering the password and a button to submit the form. The UI operations involve getting data from each field.
- There is another UI class that is of ForgotPasswordUI it has a field to input text and a submit button. The operation possible are to get the email from the input field.
- Final UI class is of ResetPasswordUI which has 2 input fields and a submit button both input fields to enter new password. The operation defined are getters for both of the fields.
- User Class has two private class attributes one is email and other is password and both the attributes have getters and setters.
- The Login class has a public attribute that is errMsg used to store error during the login process. The operations defined are the validation of the email and password and the user (once validation is done the user is logged in) and a getter for the error.
- SignUp class has a public attribute errMsg and has similar functionality implemented is validation of email password and check for the both of the password entries are the same and then the user is created using another function and then there is a getter for the error message if occurs during the registration process.
- ForgotPassword has an errMsg as public attribute a validator that checks whether the email is registered or not. A function sends a generated link to user on the submitted email and there is a setter for the Error message.
- The last class is the control component it loads all the UI components and sets errors in the UI if any. Gets user from the user class





## Safe Home Security Function Component by Team-1

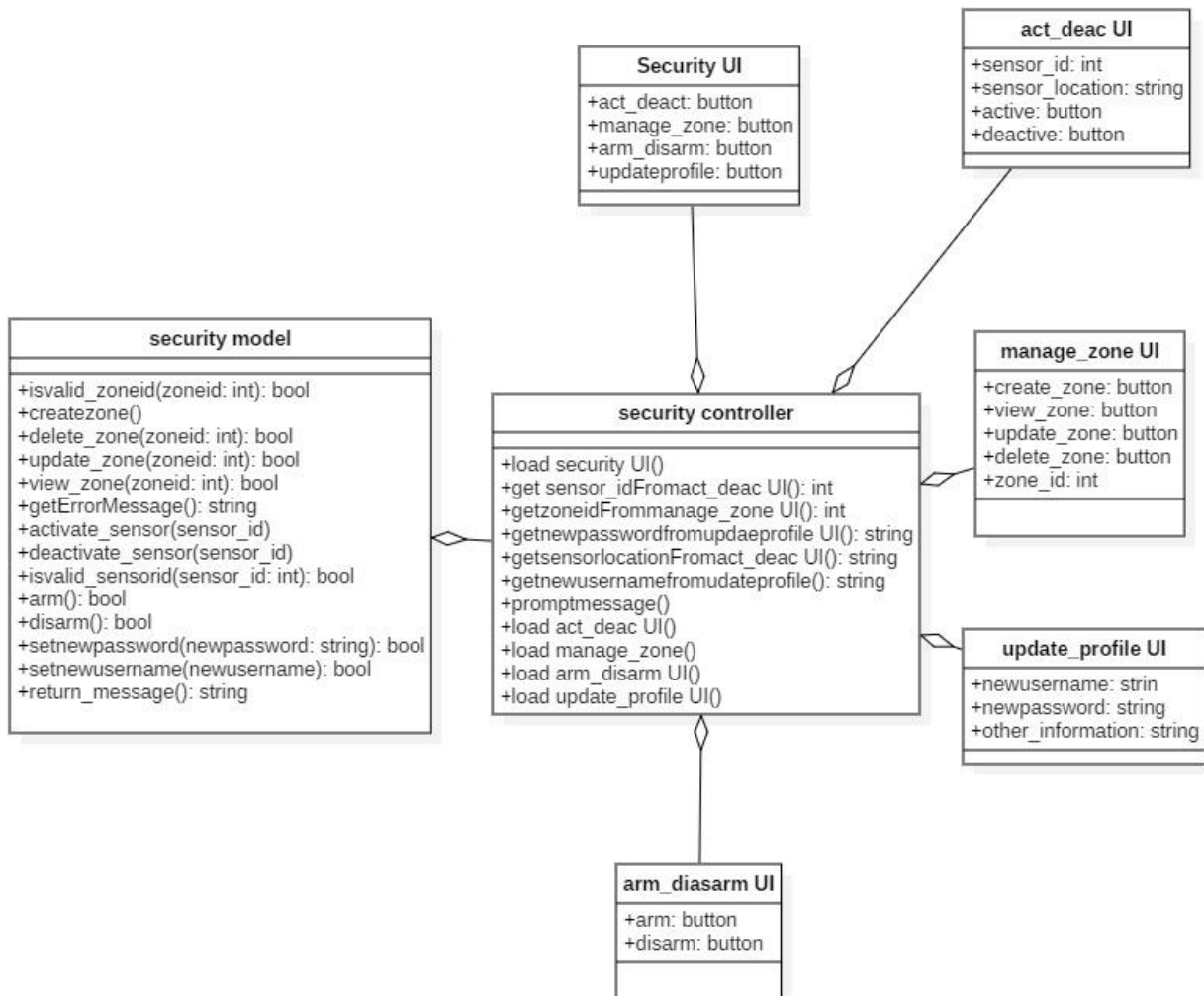


Figure 3: Safe Home Security Function



## Description Safe Home Security Function

All UI class are view classes which attract directly with user. The View presents the model's data to the use

### 1. Security UI

This is the class which is load by controller first and from that class user can select which operation which he want to do through buttons.

### 2. act\_deac UI

This class which is working as view for activate and deactivate the sensors. In this class user can enter sensor id, sensor location and sensor type of that sensor which he want to activate and deactivate.

### 3. manage\_zone UI

This class is also working as view for managing security zones in this user can select whether he will create a zone, delete a zone, update a zone and view a zone through buttons and this will also get zone id of that zone on which user can apply these operations which are discussed above.

### 4. update\_profile UI

This is also a other view class diagram in which user can update his profile. User can enter new password or new username. Other information can also be updated.

### 5. arm\_diasarm UI

This is also a view class where user can select arm the system or disarm the system through buttons.

### 6. security controller

The security controller class which act as controller. The controller translates the user's interactions with the view into actions that the model will perform.

- **Load security UI()**  
This function loads the security UI.
- **load act\_deac UI()**  
This function loads the act\_deac UI.
- **load manage\_zone()**  
This function loads the manage\_zone UI.
- **load arm\_disarm UI()**  
This function loads the arm\_disarm UI.
- **load update\_profile UI()**  
This function loads the update\_profile UI.
- **get sensor\_idFromact\_deac UI()**  
This function get the sensor id of data type int from act\_deac sensor UI.
- **getzoneidFrommanage\_zone UI()**  
This function gets zone id of data type int from manage\_zone UI.
- **getnewpasswordfromupdaeprofile UI()**  
This function will get new password of data type string from update profile UI.
- **getsensorlocationFromact\_deac UI()**  
This function will get sensor location of data type string from act\_deact UI.
- **getnewusernamefromudateprofile()**







This function will get new username of data type of string from update profile UI.

- **promptmessage()**

This function will prompt/show the message receive from model class.

## 7. security model

This is the model class which perform all the backend logic or functions which user ask or define to task. Controller class send information or call the functions of model class to perform the work of user. Model class cannot connect with view class directly but indirectly through controller class.

- **isvalid\_zoneid(zoneid:int)**

This function is used to check that is there zone exist on given zoneid. It takes zoneid as parameter and returns the true (exist) or false (not exist)

- **createzone()**

This function will create the new security zone and increment in the zoneid directly.

- **delete\_zone(zoneid:int)**

This function is used to delete the existing security zone in the system. It takes the zoneid as parameter of that zone which has to be deleted. It returns the true (deleted) or false (not deleted).

- **update\_zone(zoneid:int)**

This function is used to update the existing security zone in the system. It takes the zoneid as a parameter of that zone which has to be updated. It returns the true (updated) or false (not updated).

- **view\_zone(zoneid:int)**

This function is used to view the existing security zone in the system. It takes the zoneid as a parameter of that zone which we want to view. It returns the true (viewed) or false (not viewed).

- **activate\_sensor(sensor\_id:int)**

This function will activate the sensor of given sensorid. It takes the sensor\_id as parameter. This sensor return the bool value.

- **deactivate\_sensor(sensor\_id:int)**

This function will deactivate the sensor of given sensorid. It takes the sensor\_id as parameter. This sensor return the bool value.

- **isvalid\_sensorid(sensor\_id:int)**

- This function is used to check that is there sensor exist on given sensorid. It takes sensorid as parameter and returns the true (exist) or false (not exist)

- **arm()**

This function is used to arm the system.

- **disarm()**

This function is used to disarm the system.

- **setnewpassword(newpassword:string)**

This function is used to update the userprofile through setting new password. This function will take the newpassword as parameter and update the password.

- **setnewusername(newusername:string)**





This function is used to update the userprofile through setting new user name. This function will take the newusername as parameter and update the username.

- **return\_message()**

This function will return the error messages to controller class.