Product Design Document

Team <21, DASS-21, Archit Goyal, Akash Verma, Priyanshu Madaan, Vishal Verma>

Design Overview

Architectural design :

The end user can use their mobile phone to log into the system an initial check is performed for whether the hardware device is ON or not. Only if the Hardware is authorized and ON then the user is authenticated. Once the authentication's done successfully the user is able to send the control signals to the Hardware machine. At the hardware machine the driver program will continuously track for the change in the status & will accordingly send the signals to the Circuit. When a user select a change in the status for any of the device i.e ON the data from the hand held is sent to the Web Server in a string format, where the Website is hosted. On the server the status is stored in the database in their respective device field. is used to retrieve the status of the devices in a timely pattern for every 10sec. These changes come in to form of cookies which are temporary internet files from the web server& are stored on the computer in the name of the web site every 10 sec as the page refreshes the new cookie values are updated.

System interfaces

User Interface:

Android app based user interface is required for controlling the classroom.

API' s:

API' s are not yet created, but considering a decoupled model.

API	*TYPE*	*JSON*
/classroom/currentStats	*GET*	*{ inTemp=float, outTemp=float, inHumidity=int, outHumidity=int, co2=int, power=float }*
/classroom/updateStats	POST	{ inTemp=float, outTemp=float, inHumidity=int, outHumidity=int, co2=int, power=float }
/ac<ac_no>/current_stats	GET	{ acNo=int, temp=int, mode=string, swing=*<on/off>* }
/ac<ac_no>/set_temp/<temp>	*GET*	

/ac<ac_no>/set_mode/<mode (DRY/COOL/FAN) >	*GET*	
/ac<ac_no>/set_swing/<(ON/OFF) >	*GET*	
/ac<ac_no>/power/<(ON/OFF) >	*GET*	
/projector<projector_no>/current_stats	*GET*	*{ projectorNo=int , input = string , aspectRatio=string }*
/projector<projector_no>/power/<(ON/OFF)>	*GET*	
/projector<projector_no>/input/<input_name>	*GET*	
/projector<projector_no>/aspectRatio/<aspect_ratio>	*GET*	
/lights/activate	*POST*	*{ current_light_configuration=string }*
/lights/current_config	*GET*	*{ current_light_configuration=string }*
*/event/*add	POST	{ name=string , projector=(ON/OFF), boardLight=(ON/OFF), audienceLights=(ON/OFF), date=date, timeSlot=int , occupancy=int }
/event/get_current/<slot>	GET	{ name=string , projector=(ON/OFF), boardLight=(ON/OFF), audienceLights=(ON/OFF), date=date, timeSlot=int , occupancy=int }
/event/validateDateTime	POST	{ date=date, timeSlot=int }

CONSTRAINTS OR NOTES

Response codes

Ok = 200

Not Ok = 204

Response codes should be appended to all responses

validateDateTime returns 200 iff Date and slot are unique i.e. no entry with the given slot on given date exists

ac_no = {1,2,3,4} for 4 different AC

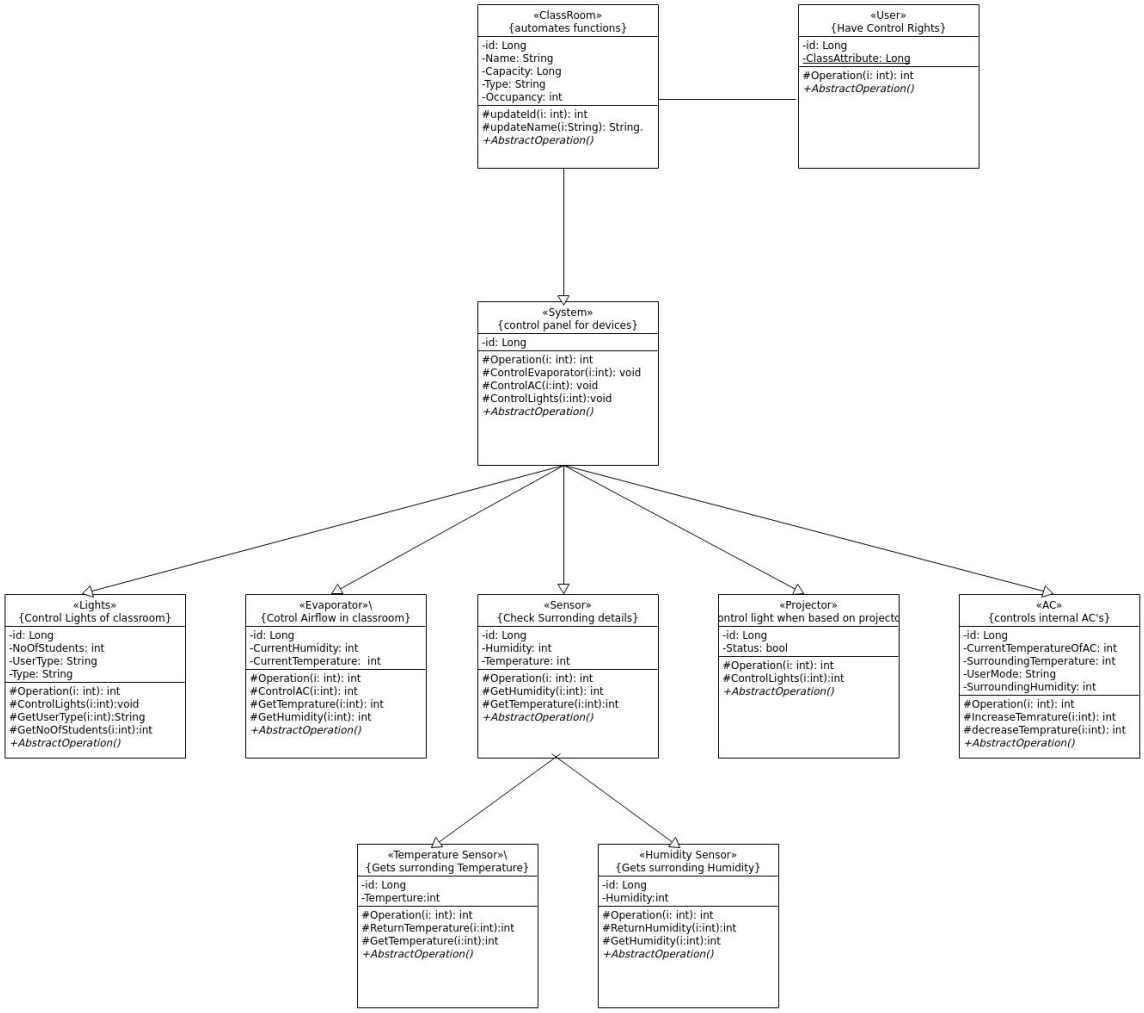
projector_no= {1,2}

(ON/OFF) need to store one of the strings either ON or OFF and correspondingly turn ON and OFF the equipment

<place_holder> are placeholder and the actual api should not have '<' or '>'

/event/get_current/slot will return JSON of todays date and with given slot and <slot> will be an integer no

Model :



URL : https://gitlab.com/dass-2020/dass21/blob/developer/diagrams/UML_Class.png

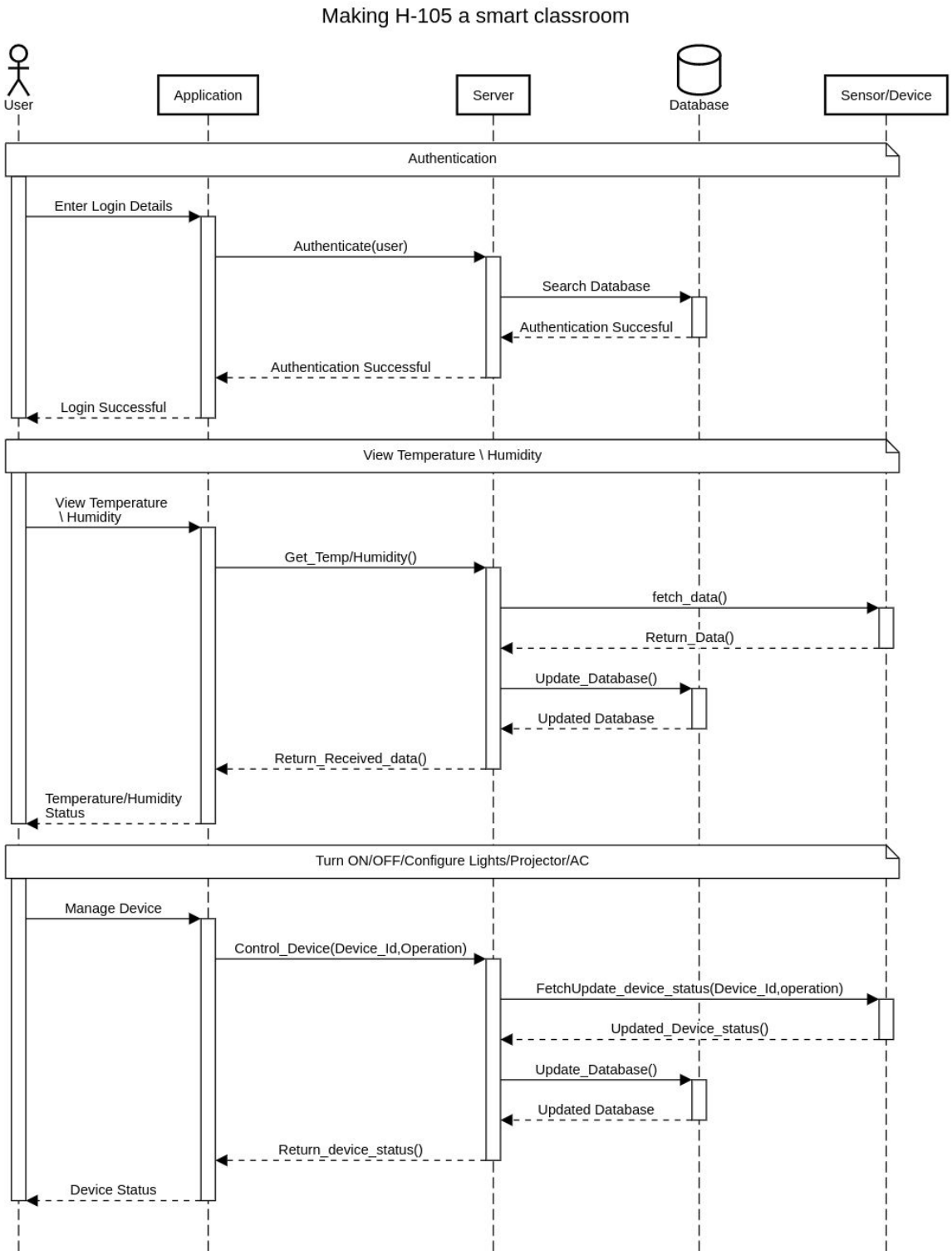
Classroom	Class state : *id : Long* *name : String* *capacity : Long* *occupancy : int* Class behavior : *updateid(I : int) : int* *updateName(I : string) : string* *AbstractOperation()*
System	Class state : *id : Long* Class behavior : *Operation(i : int) : int* *ControlEvaporator(i : int) : void* *ControlAC(i : int) : void* *ControlLights(i : int) : void*

	AbstractOperation()
Lights	Class state : *id : Long* *NoOfStudents: int* *UserType: String* *Type: String* Class behavior : *Operation(i : int) : int* *ControlLights(i : int) : void* *GetUserType(i : int) : String* *GetNoOfStudents(i : int) : int* *AbstractOperation()*
Evaporator	Class state : *id : Long* *CurrentHumidity : int* *CurrentTemperature : int* *Class behavior :* *GetTemprature(i : int): int* *Operation(i : int) : int* *ControlAC(i : int) : int* *GetHumidity(i : int) : int* *AbstractOperation()*
AC	Class state : *id : Long* *CurrentTemperatureOfAC : int* *UserMode : int* *SurroundingHumidity : int* Class behavior : *Operation(i : int) : int* *IncreaseTemperature(i : int) : int* *decreaseTemperature(i : int) : int* *AbstractOperation()*
Projector	Class state : *id : Long* *Status : bool* Class behavior : *Operation(i : int) : int* *ControlLights(i : int) : int* *AbstractOperation()*
Sensor	Class state : *id : Long* *Humidity : int* *Temperature : int* Class behavior : *Operation(i : int) : int* *GetHumidity(i : int) : int* *GetTemprature(i : int) : int* *AbstractOperation()*
Temperature Sensor	Class state : *id : Long* *Temperature : int* Class behavior : *Operation(i : int) : int* *ReturnTemperature(i : int) : int* *GetTemperature(i : int) : int* *AbstractOperation()*

Humidity Sensor	Class state : *id : Long* *Humidity : int* Class behavior : *Operation(i : int) : int* *ReturnHumidity(i : int) : int* *GetHumidity(i : int) : int* *AbstractOperation()*
Users	Class state : *id : Long* *ClassAttribute : Long* Class behavior : *Operation(i : int) : int* *AbstractOperation()*

.

Sequence Diagram(s)



URL : https://gitlab.com/dass-2020/dass21/blob/developer/diagrams/UML_Sequence.svg

Design Rationale

The Internet of Things, IoT, is in a huge way and people are rapidly inventing new gadgets that enhances lives. The price of microcontrollers with the ability to talk over a network keeps dropping and developers can now tinker and build things inexpensively. IoT based home automation project is done using low cost ESP8266 Espino ESP-12 WiFi Module, It uses relays and few simple components, four electrical devices can be controlled and temperature can be monitored . ESP-12 is low cost module is used here. Homes of the 21st century will become more and more self - controlled and automated due to the comfort it provides, especially when employed in a private home. A home automation system is a means that allow users to control electric appliances of varying kind