HueEmulator(User Interface)
- + signIn()
- + signUp()
- + HueEmulator:Bridge

Signup
- -invalidMessage:Wt:Wtext
- -fname: Wt:WLabel
- - firstName: Wt:WLineEdit
- lname: Wt::WLabel
- - lastName: Wt::WLineEdit
- - uName: Wt::WLabel
- - username:Wt::WLineEdit
- - pass: Wt::WLabel
- - password: Wt::WLabel
- - Cpass: Wt::WLabel
- - confirmpass: Wt::WLineEdit
- -signin: Wt::WPushButton
- - signup:Wt:WPushButton
- ---------------------------------------
- + showDialog:
- + onClick:
- + invalid:

Sigin
- - dialog:Wt::WDialog
- - email:string
- - displayEmail:Wt:WText
- - invalidMessage: Wt:WTest
- - uName: Wt::WLabel
- - user: Wt::WLineEdit
- - pass:Wt::WLabel
- - password: Wt::WLineEdit
- - siginButton: Wt::WPushButton
- - signInButton: Wt::WPushButton
- ---------------------------------------
- + showDialog:
- + onClick:
- + invalid:
- + setPointer:

hashFunction
- - addString: String
- - length: String
- - tempValues: vector<string>
- - finalValues:vector<string>
- ---------------------------------------
- + hashFunction:
- + getHash: String

ChangeLights: Wt::WContainerWidget
- + ChangeLights(ip, port, user)
- + setLightName(ip, port, user)
- + getLights(): void
- + getAllLights(): void
- + setLight(ip, port, user)
- + handle: void
- ---------------------------------------
- - *on: Wt:WLineEdit
- - * sat: Wt::WLineEdit
- - * hue: Wt::WLineEdit
- - *bri: Wt::WLineEdit
- - transaction: Wt::WLineEdit
- - *effect: Wt::WLineEdit
- - *lightNum: Wt::WLineEdit
- - *button: Wt::WPushButton
- - *lightName: Wt::WLineEdit
- - *getLightResult: Wt::WText

Authenticate
- - userName: string
- - password: Hash
- ---------------------------------------
- + Authenticate(userName, password)
- + getHash
- + getName

HueEmulator(Lights)
- - brightness: int
- - colors: (hexidecimal)
- - status: int
- - effect: int
- - transition:int

groupContainer:WContainerWidget
- +GroupContainer(*ip, *port, *user)
- + handle: void
- + createGroup(*ip, *port, *user)
- + getGroup: void
- + setGroup(*ip, *port, *user)
- + deleteGroup(*ip, *port, *user)
- ---------------------------------------
- - *on3: Wt::WLineEdit
- - *hue3: Wt::WLineEdit
- - *bri3: Wt::WLineEdit
- - transition3: Wt::WLineEdit
- - *effect: Wt::WLineEdit
- - groupgetResult: Wt::WText
- - * groupName: Wt::WLineEdit

HueEmulator(Bridge)
- - lightNum: Wt::WLineEdit
- - on:Wt::WLineEdit
- - sat:Wt::WLineEdit
- - hue:Wt::WLineEdit
- - bri:Wt::WLineEdit
- - transition:Wt::WLineEdit
- - effect: Wt::WLineEdit
- - lightget:Wt::WLineEdit
- - groupget:Wt::WLineEdit
- - result: Wt::WText
- - lightgetResult: Wt::WText
- - groupgetResult: Wt::WText
- ---------------------------------------
- + HueEmulator();
- + ~HueEmulator:
- + editBridge:Bridge(editData)
- + removeBridge: RemoveBridge(bridgeNam
- + addLight: AddLight(name, ip, port, user)
- + getGroupVector:Vector<Group>
- + changeLight(ip, port, user)
- + getLightAtIndex:Light(index)
- + removeLight:Light(index)
- + createGroup:Group(ip, port, user)
- * createSchedule()
- + deleteBridge()

EditBridge: Wt::Dilalog
- + showDialogBox:void
- + onClick: void
- + inputChecker: bool
- ---------------------------------------
- -userEmail: string
- - currentBridge:string
- - bridgName: string
- - bridgeList: Wt::WComboBox
- - invalidMessage: Wt::WText
- - *bName: Wt::WLabel
- - *bridgeName:Wt::WLineEdit
- - *bPort: Wt:WLabel
- - *bridgePort: Wt::WLineEdit
- - *bLocation: Wt::WLabel
- - *bridgeLocation: Wt::WLineEdit
- - *bUser : Wt::WLabel
- - *bridgeUser: Wt::WLineEdit

RemoveBridge:Wt::WDialog
- +showDialog: void
- + onClick: void
- ---------------------------------------
- - buttonObject: vector<WPushButto
- - userEmail: string
- - bridgeList: Wt::WComboBox
- - currentBridge: string
- - bridgName: string

AddBridge : Wt::WDialog
- + showDialogBox:void
- + onClick: void
- + addHandler: void
- + inputHandler: bool
- ---------------------------------------
- - name: string
- - ipAddress: string
- - port: string
- - user: string
- - bridgeList: Wt::WComboBox
- - invalidMessage: Wt::WText
- - *bName: Wt::WLabel
- - *bridgeName:Wt::WLineEdit
- - *bPort: Wt:WLabel
- - *bridgePort: Wt::WLineEdit
- - *bLocation: Wt::WLabel
- - *bridgeLocation: Wt::WLineEdit
- - *bUser : Wt::WLabel
- - *bridgeUser: Wt::WLineEdit

Bridge(Persistece)
- - email: string
- - firstName: String
- - lastName: String
- - password: String
- - Bridge: Bridge[]
- ---------------------------------------
- + getPassword:Hash
- + createAccount: (first, last, pword)
- + authenticate:(username, pword)

ScheduleContainer: Wt::WContainerWidget
- + ScheduleContainer()
- + handle: void
- + createSchedule: void
- + getSchedule: void
- + setSchedule: void
- + deleteSchedule: void
- ---------------------------------------
- - *on1: Wt::WLineEdit
- - *sat1: Wt::WLineEdit
- - *hue1: Wt::WLineEdit
- - *bri1: Wt::WLineEdit
- - *transition1: Wt::WLineEdit
- - *effect1 : Wt::WLineEdit
- - *LorG: Wt::WLineEdit
- - *LorG1: Wt::WLineEdit
- - *scheduleName: Wt::WLineEdit
- - *description: Wt::WLineEdit
- - *scheduleNum: Wt::WLineEdit
- - *lightNum2: Wt::WLineEdit
- - *groupNum2: Wt::WLineEdit
- - *on2: Wt::WLineEdit
- - *hue2: Wt::WLineEdit
- - *bri2: Wt::WLineEdit
- - *transition2:Wt::WLineEdit
- - *effect2: Wt::WLineEdit
- - *localtime: Wt::WLineEdit
- - *lightgetResult: Wt::WText