

<<data abstraction>> ControlTemp
currentTemp: int desiredTemp: int heaterStatus: char acStatus: char tempTimeLog: datetime
readCurrentTemp(in currentTemp) reachDesiredTemp(in readCurrentTemp(in currentTemp), in heaterStatus, in acStatus, out tempTimeLog)) maintainDesiredTemp((in readCurrentTemp(in currentTemp), in heaterStatus, in acStatus, out tempTimeLog))

<<data abstraction>> Set Desired Temp
motionDetected: char recordtime: datetime systemMessage: string currentTemp: int comfortableTemp: int safeTemp:int
readMotion(in motionDeteced) sendSystemMessage(out systemMessage) setDesiredTemp(in currentTemp, out comfortableTemp, out recordTime) setSafeTemp(in motionDetected, out safeTemp, out recordTime)

<<data abstraction>> Record Automobile Gas Use
isEngineOff: char lastGasLevelUsed: int currentGasLevelUsed: int totalUsage: int recordGas: int recordMileage: double
isCarOff(out isEngineOff) calculateGasUse(in lastGasLevelUsed, in currentGasLevelUsed, out totalUasge) recordInfo(in recordGas, in recordMileage)

<<data abstraction>> Light On/Off
isSwitchOn: char motionDetected: char turnLightsInRoom: char
lightFlippedOn(in isSwitchOn) detectMotion(in isSwitchOn, in motionDetected, out turnLightsInRoom) lightFlippedOff(in isSwitchOn)

<<data abstraction>> Monitor Water Leak
isWaterLeak: char notifyResident: string recordTime: datetime
isThereALeak(in isWaterLeak) sendResidentText(out notifyResident) recordLeakTime (in recordTime)

<<data abstraction>> Detect Fire
timeLog: datetime alarmTimeLog: datetime COTimeLog: datetime isFireAlarmActive: char currentCOLevels: int isSmokeDetected: char isSprinklerOn: char sendResidentNotification: string systemDetectsFire: char isSprinklerActive: char alertFireStation: string
detectsSmokeOrFire(in isSmokeDetected) startFireProtocol(out sendResidentNotification, in isFirealarmActive) recordAlarmTime(in alarmTimeLog) highCODetected(in currentCOLevels, out isSprinklerOn, in COTimeLog) activateSprinkler(in isSprinklerOn, in systemDetectsFire, out isSprinklerActive) notifyFireStation(out COTimeLog, out alertFireStation) timeToNotify (in timeLog)

<<data abstraction>> Home Security
imageURL: string isSuspiciousPerson: char isPersonBreakingIn: char notifyResident: string recordTime: datetime isAlarmActive: char notifyResident: string notifyPolice: string
storeAndAnalyzeImages(in imageURL) suspicionDetected(storeAndAnalyzeImages(in imageURL), out isSuspiciousPerson, out notifyResident) breakInOccurs(storeAndAnalyzeImages(in imageURL), in isAlarmActive, out notifyResident, out notifyPolice)

<<data abstraction>> Login
username: string password: string Credentials: string isValid: char recordTime: datetime welcomeMessage: string errorMessage: string
enterCredentials(in username, in password) validateCredentials(in Credentials, in isValid) credentialsValid(in recordTime, out welcomeMessage) credentialsInvalid(in recordTime, out errorMessage)

<<data abstraction>> View/Change Values
isViewSelected: char isEditSelected: char isLogoutSelected: char newSafeTemplInput: int newComfortableTemplInput: int recordTime: datetime loggedOut: char
selectsView(in isViewSelected) selectsEdit(in isEditSelected) userUpdatesTemps(in newSafeTemplInput, in newComfortableTemplInput, in recordTime) userLogout(in isLogoutSelected, out loggedOut)

<<data abstraction>> Record Electricity Use and Bill
Timer: double electrcityUsed: int electricityBill: double sendToEC: string sendToLC: string isBillValidEC: char sendValidBillEC: string sendToBlockchainLedgers: string sendInvalidBillMessageEC: string
timerAlert(out Timer, out electricityUse, out electricityBill) calculateBill (in electrcityUse, electricityBill) sendToECAndLC(out sendToEC, out sendToLC) verifyBill(in sendToEC, out isBillValidEC,) sendAValidBill(out isBillValidEC, out sendValidBillEC) sendValidBillToBlockchainLedgers(out sendToBlockchainLedgers) sendAInvalidBillMessage(out isBillValidEC, out sendInvalidBillMessageEC)

<<data abstraction>> View Electricity Use and Bill
isViewSelected: char usageAndBillPeriod: string
electrcityView(in isViewSelected) systemReadsDataFromBlockchain(in UsageAndBillPeriod) displayInfo(out usageAndBillPeriod)