SmartHub Project Design Document
Software Engineering 330

Jackson Player & Andrew Yang

Contents

Description

This application allows a user to register and login to a smart home hub. The smart home hub gives access to a smart plug, thermostat, camera and smart light. The smart plug can be turned on and off. The thermostat can have its temperature adjusted and can be switched on and off. The camera has limited space and the recording function can be turned on and off. The camera also has the ability to stream a video, by using the 'streaming' button. This video can be played and paused. Finally the smart light can be toggled as well.

There is also an administration panel. The administration panel allows a user to add or remove devices from the smarthub.

Current Implementation

We have implemented the login and registration system, along with the full UI. The full functionality has been implemented apart from the concurrent logging, which we could not get done in time due to time restraints.

To run through login screen please use "User" for the userID and "password" for the password.

Our implementation used JavaFX with fxml files for the views.

MVC

Models: CameraModel, SmartPlugModel, LightbulbModel, Thermostat, UserModelModel, TemperatureModel, DeviceModel

Views: AddDevices, AdminPanelView, AdminHomeView, CameraStreamingView, CameraView, LightbulbView, ThermostatView, SmartplugView, LoginView, RegisterView, NotifyDevicesView, NotifyUsersView, HubHomeView, RemoveDevicesView, RemoveUsersView,

Controls: LoginController, RegisterController, HubHomeController, CameraController, LightbulbController, ThermostatController, SmartplugController, NotifyUsersController, NotifyDevicesController, MediaStreamController, AddDeviceController, RemoveDeviceController, RemoveUserController,

Updates

Since assignment 3 we have implemented the device list and user list that can be modified through the administration panel. We have also implemented the media streaming capability through the camera. The design document has also been updated to reflect the latest changes with the MVC.

A variety of user interface tests and unit tests have been added for the changes we have made.

UML Diagram

The UML diagram is on the next page. If it is too difficult to read, please visit this

link for a high quality version: https://ibb.co/dJM53s2. The classes in purple are the ones added since assignment 3.

Views and User Flow

LoginView
- login: TextField
- password: TextField
- registerButton: Button
- submitButton: Button

RegisterView
- userid: TextField
- select_password: PasswordField
- re-enter-pass: PasswordField
- adminRadioButton: RadioButton
- userRadioButton: RadioButton
- submitButton: Button

AdminHomeView
- adminButton: Button
- deviceList: ListView
- shutdownButton: Button

HubHomeView
- deviceList: ListView
- shutdownButton: Button

AdminPanelView
- goToHubButton: Button
- addDevice: Button
- removeDevice: button
- addUser: Button
- removeUser: Button
- notifyDevice: Button
- notifyUsers:Button

LightbulbControlView
- statusLabel: Label
- toggleButton: Button

ThermostatControlView
- statusLabel: Label
- temperatureLabel: Label
- startButton: Button
- stopButton: Button
- backButton: Button

SmartPlugControlView
- statusLabel: Label
- toggleButton: Button

CameraView
- streamingButton: Button
- backButton: Button
- toggleButton: Button
- startButton: Button
- stopButton: Button

NotifyUsersView
- cancelButton: Button
- submitButton: Button
- notifyAllUsers: Button
- userList: ListView

NotifyDevicesView
- cancelButton: Button
- submitButton: Button
- notifyAllDevices: Button
- deviceList: ListView

MediaStreamView
- playButton: Button
- pauseButton: Button
- backButton: Button
- mediaStream: MediaViewer

Notifies | Updates

Controllers

SmartPlugController
- model: SmartPlug

onBackClick(ActionEvent): void
onToggleClick(ActionEvent): void

CameraController
- model: Camera
- statusLabel: Label
- recordingLabel: Label

onStopRecordClick(ActionEvent): void
onBackClick(ActionEvent): void
onStartRecordClick(ActionEvent): void

HomeController
- exitButton: Button
- model: Hub

onSelectionClick: Button

AdminHomeController
- model: DeviceModel
- registerDeviceList: ListView

onSelectClick(ActionEvent): void
onAddUser(ActionEvent): void
onShutDownClick(ActionEvent): void
initialize(): void

AdminPanelController
onAddDevice(ActionEvent): void
onAddUser(ActionEvent): void
onNotifyDevice(ActionEvent): void
onNotifyUser(): void
onRemoveDevice(): void
onRemove(User): void
onGoToHub(): void

ThermostatController
- model: Thermostat
- statusLabel: Label

onStopClick(ActionEvent): void
onBackClick(ActionEvent): void
onStartClick(ActionEvent): void
onToggle(ActionEvent): void
- convertTemp(Temperature): void
- temperatureLabel: Label

MediaStreamController
- model: Camera
- mediaStreamer: MediaView
- media: Media

onBackClick(ActionEvent): void
onPlayClick(ActionEvent): void
onPauseClick(ActionEvent): void

LightbulbController
- model: Lightbulb
- statusLabel: Label

onBackClick(ActionEvent): void
onToggle(ActionEvent): void

NotifyDevicesController
- model: DeviceModel
- notifyDeviceList: ListView

onCancelClick(ActionEvent): void
onNotifyAll(ActionEvent): void
onSubmitClick(ActionEvent): void
initialize(): void

NotifyUsersController
- model: UserModel
- notifyUserList: ListView

onCancelClick(ActionEvent): void
onNotifyAll(ActionEvent): void
onSubmitClick(ActionEvent): void
initialize(): void

RegisterController
- model: UserModel
- selectPassword: Label
- reenter_password: Label

registerSubmitClick(ActionEvent): void

LoginController
- model: UserModel
- selectPassword: Label
- reenter_password: Label

loginSubmitClick(ActionEvent): void
registerClick(ActionEvent): void

Notifies & Passes Info | Updates

Models

DeviceModel
- deviceList: HashMap
- typeList: HashMap

+ addDevice(Device, String,String): boolean
+ removeDevice(String): boolean
+ checkNickname(String): boolean
+ getDevList(String,String): HashMap
- JSONToMap(String): HashMap
+ updateMap(): void
+ getTypeList(String,String): HashMap

UserModel
- userid: HashMap
- accessLevel: HashMap

+ addUser(String, String,String): boolean
- userIdValid(String): boolean
- checkUserAndPassword(String,String): boolean
- checkAdmin(String,String): boolean
- JSONToMap(String): HashMap
+ updateHashMap(): void

Hub
- deviceList: HashMap
- typeList: HashMap
- isOn: boolean

+ startup(): void
+ shutdown(): void
+ register(Device): void
+ unregister(Device): void
+ getDevices(): HashMap

Temperature
- temperature: double
- Unit: Unit

+ getUnit(): Unit
+ getTemperature(): Temperature
+ setTemperature(double, Unit): void

Thermostat
- status: Status
- temperature: Temperature
- identifier: UUID
- hub: Hub

+ setStatus(Status): void
+ getStatus(): Status
+ getTemperature(): Temperature
+ setTemperature(Temperature): void

Camera
- diskSize: int
- memoryUsed: int
- MAX_DISK_SIZE: int
- hub: Hub
- isStreaming: boolean
- isRecording: boolean

+ setStatus(Status): void
+ getStatus(): Status
+ record(): void
+ getIsRecording(): boolean
+ getIsStreaming(): boolean

Lightbulb
- status: status
- isOn: boolean
- identifier: UUID
- hub: Hub

+ setStatus(Status): void
+ getStatus(): Status
+ toggle(): void
+ getIsOn(): boolean

SmartPlug
- status: status
- isOn: boolean
- identifier: UUID
- hub: Hub

+ setStatus(Status): void
+ getStatus(): Status
+ toggle(): void
+ getIsOn(): boolean

Unit <<enum>>
CELSIUS, FARENHEIT

TemperatureOutOfBoundsException <<exception>>

CameraFullException <<exception>>

Status<<enum>>
NORMAL, ALERT, WARN, ON, OFF

Uses | Throws