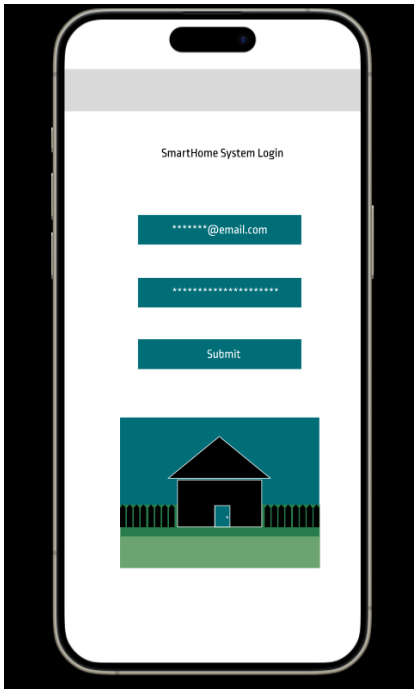
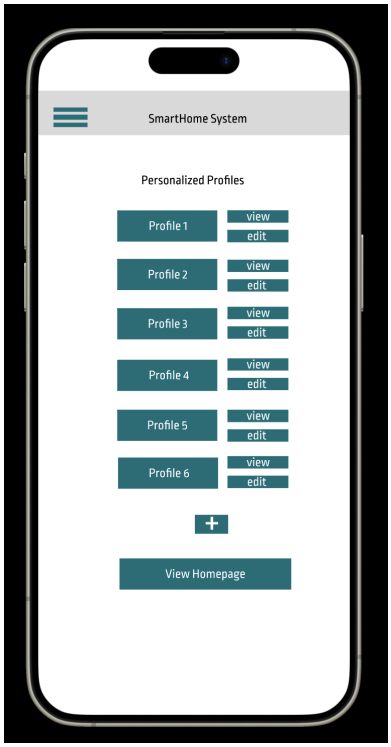
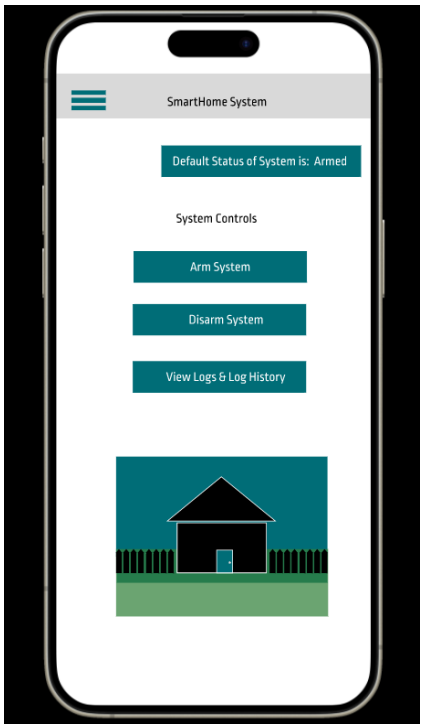
Group 6 Team Members: Vaishnavi Alavala, Isha Kandunoori, Vivien Pang, Erica Chang, Vy Dinh, Asfandiyar Khan

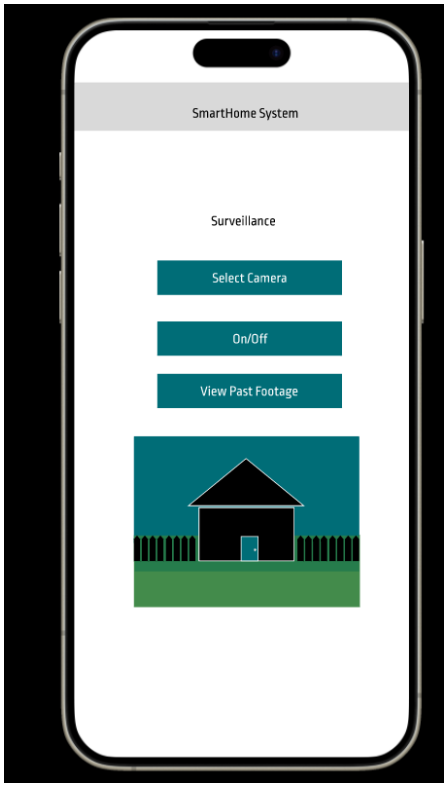


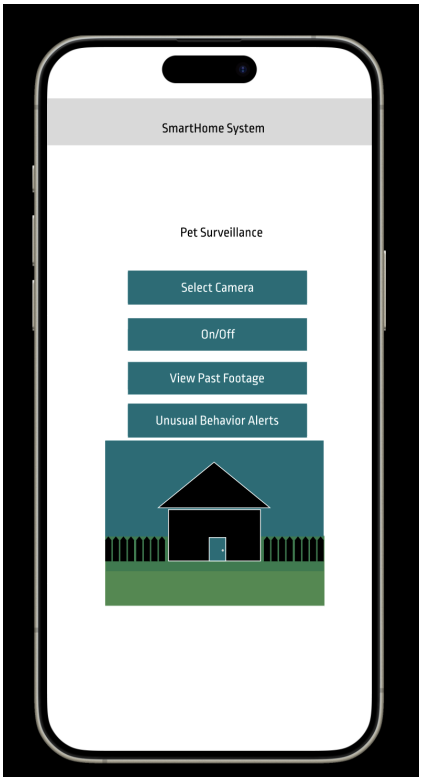


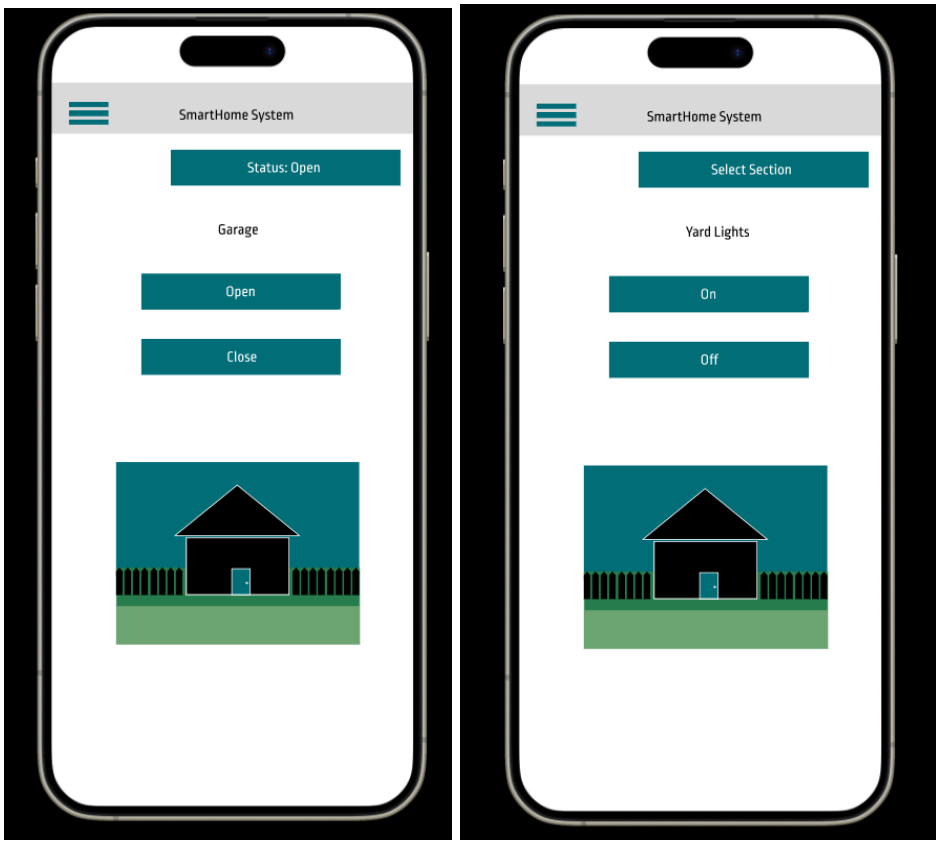


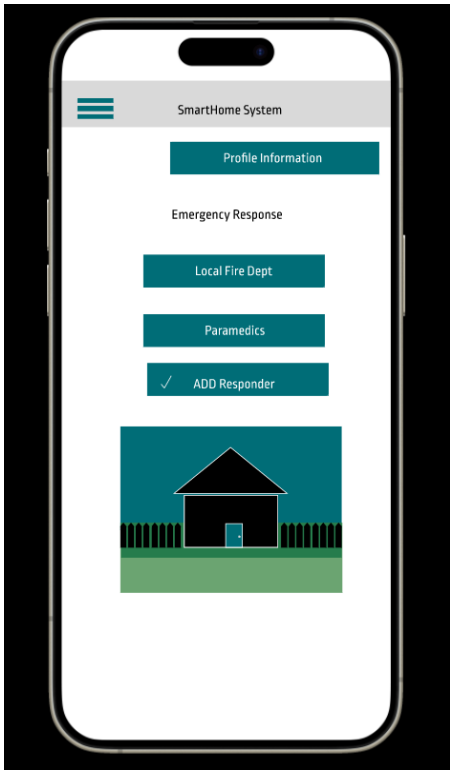

SmartHome System
Active Logs
ARM
DISARM

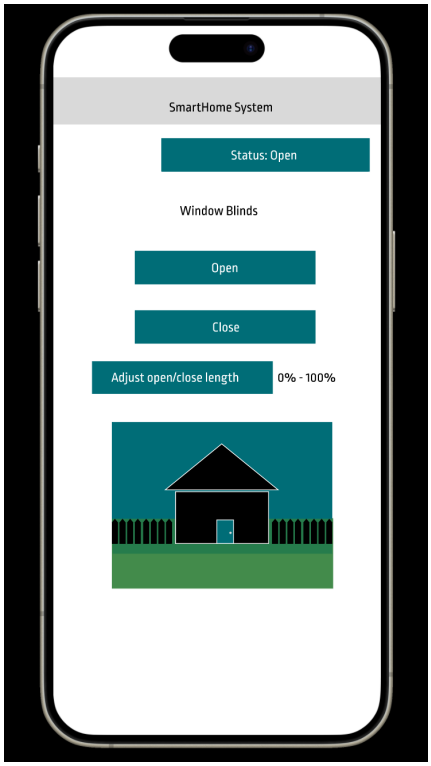

SmartHome System
Log History
Events
1. ----------
2. ----------
3. ----------
4. ----------
5. ----------
Event Description
Event Date
_/_/2023
Clear Logs
Questions? Digital Assistant

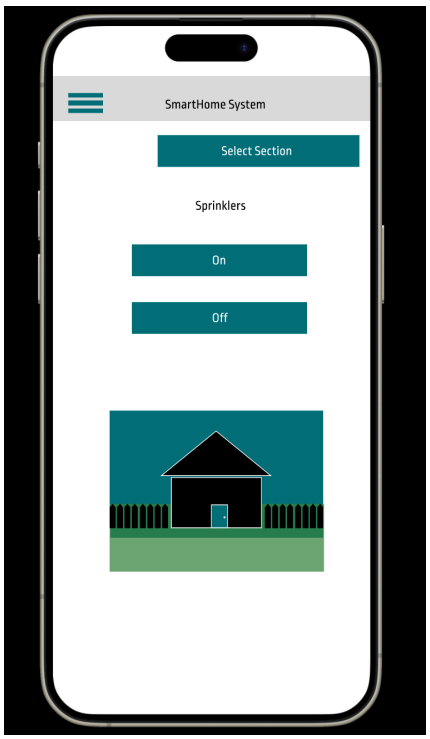

SmartHome System
Surveillance
Select Camera
On/Off
View Past Footage

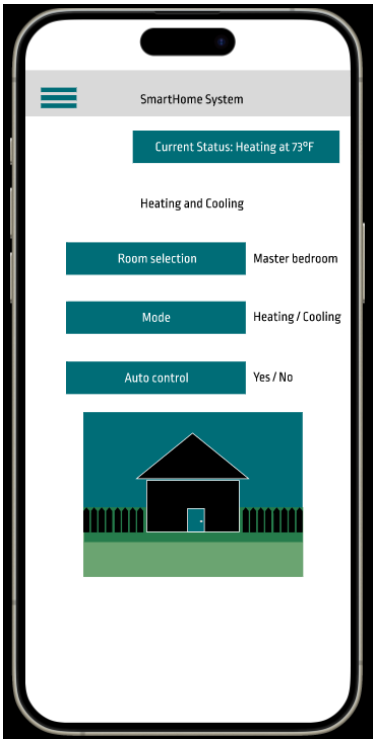

SmartHome System
Pet Surveillance
Select Camera
On/Off
View Past Footage
Unusual Behavior Alerts

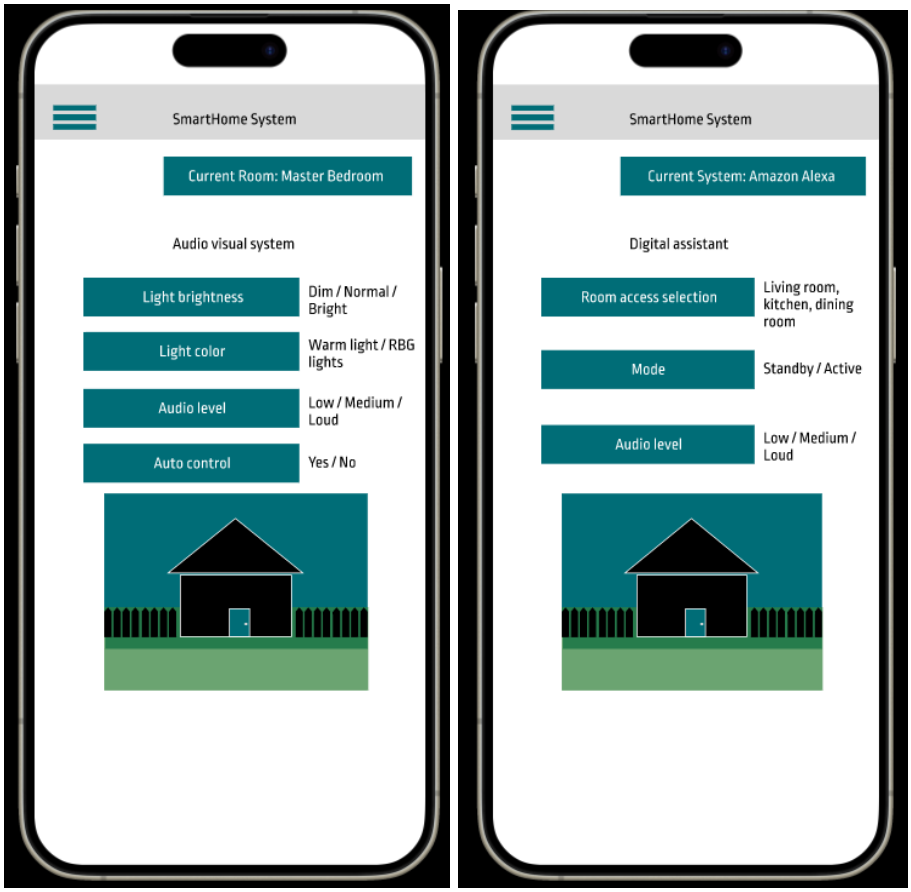

SmartHome System
Status: Open
Garage
Open
Close
SmartHome System
Select Section
Yard Lights
On
Off

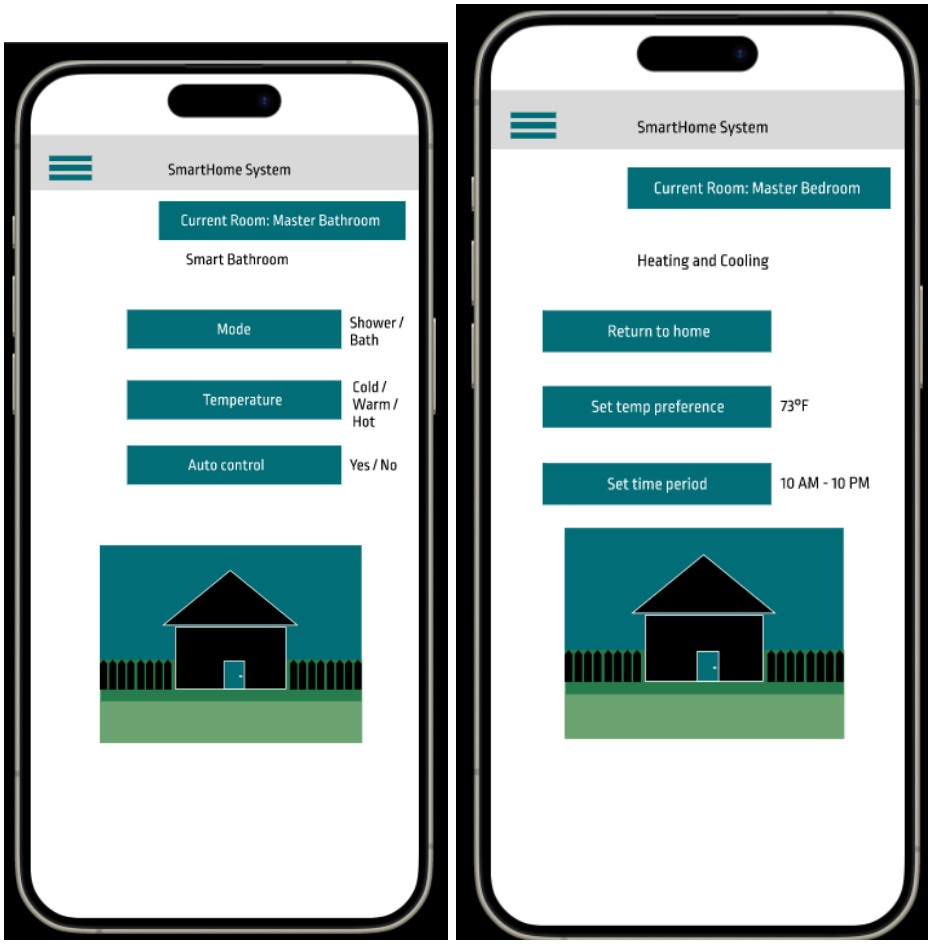

SmartHome System
Profile Information
Emergency Response
Local Fire Dept
Paramedics
ADD Responder

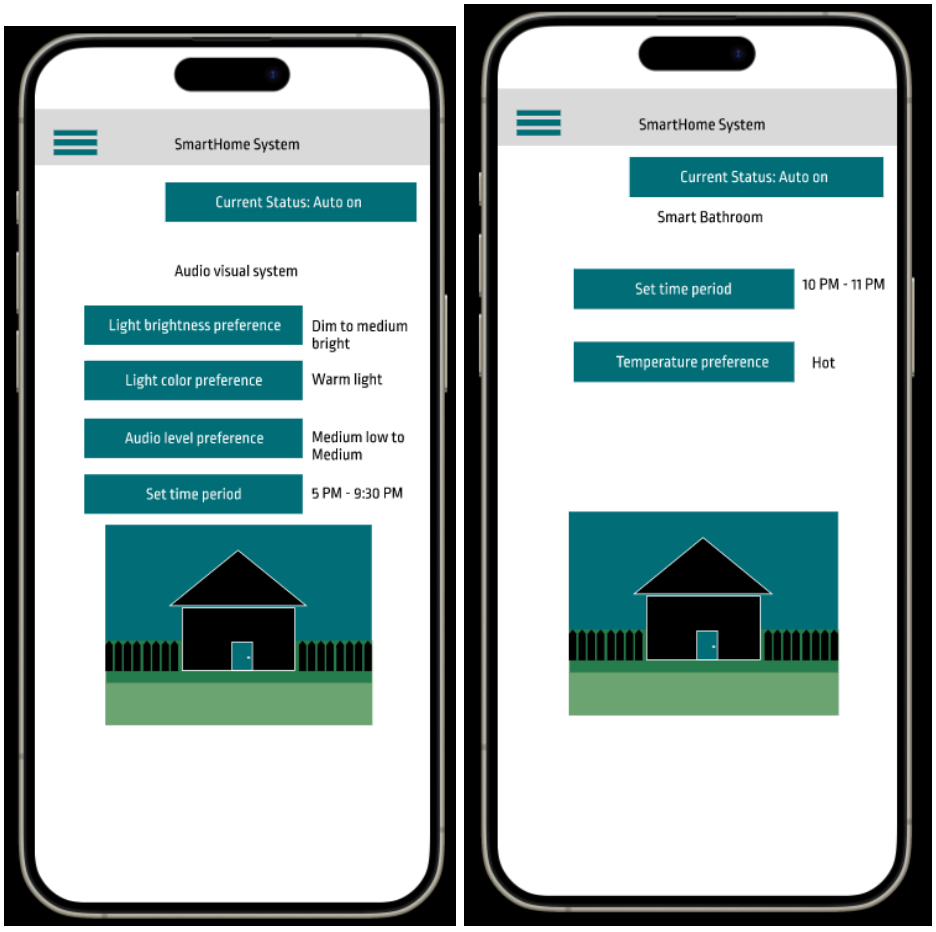

SmartHome System
Status: Open
Window Blinds
Open
Close
Adjust open/close length
0% - 100%

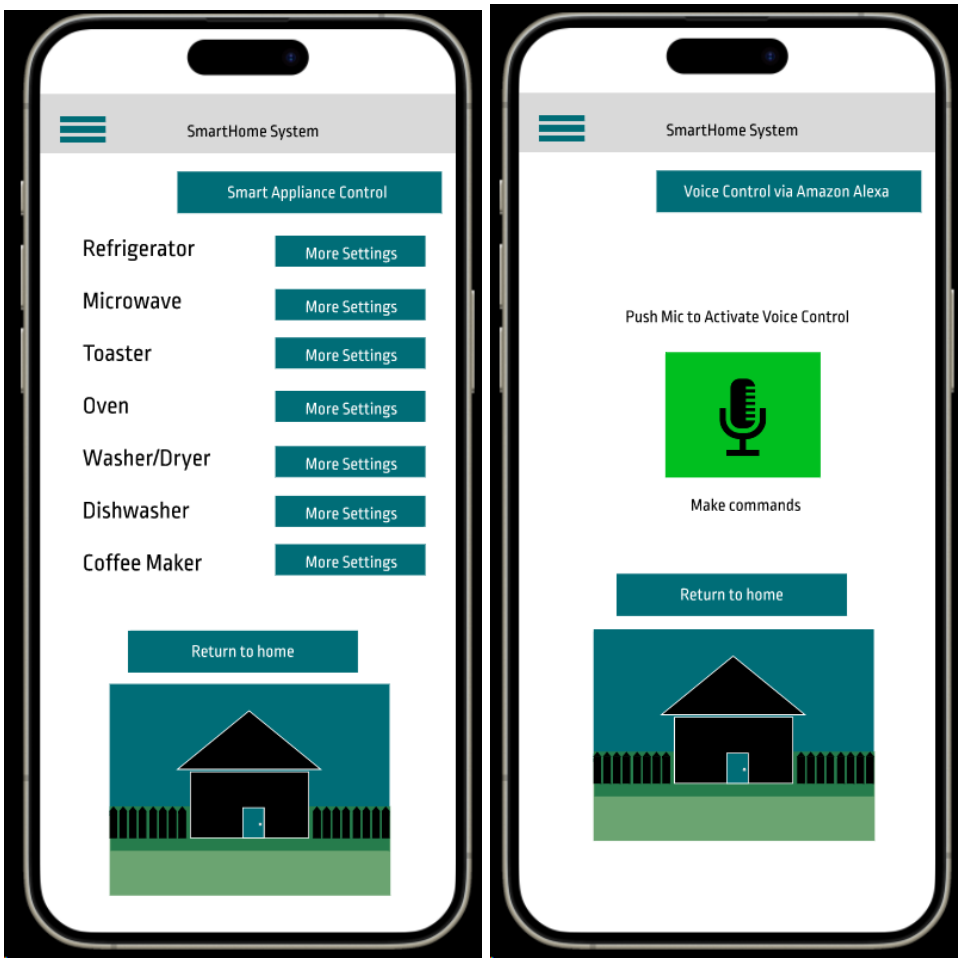

SmartHome System
Select Section
Sprinklers
On
Off

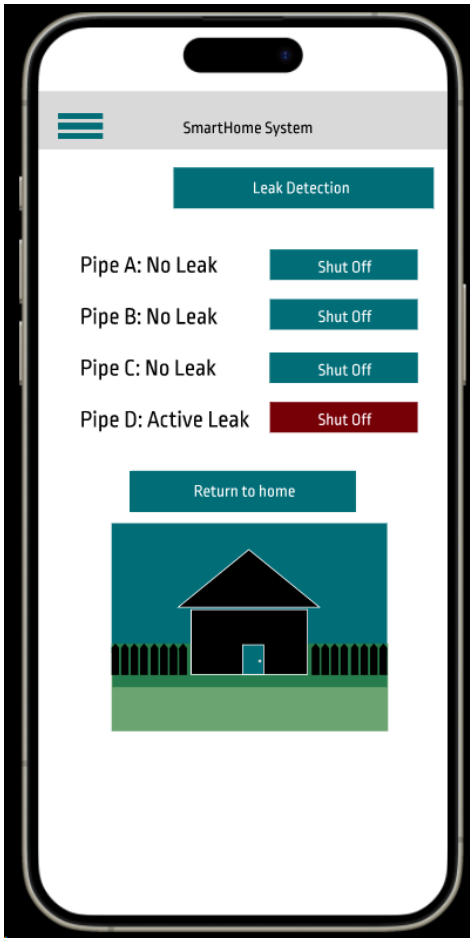

SmartHome System
Current Status: Heating at 73°F
Heating and Cooling
Room selection
Master bedroom
Mode
Heating / Cooling
Auto control
Yes / No

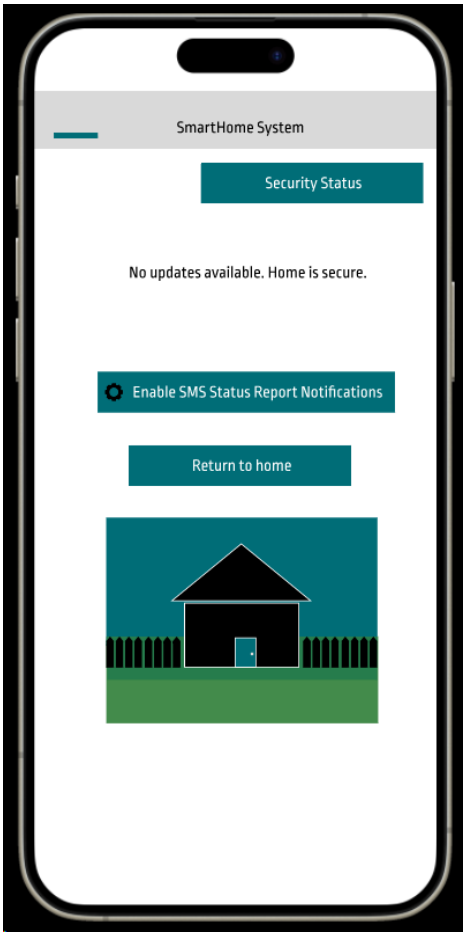

SmartHome System
Current Room: Master Bedroom
Audio visual system
Light brightness
Dim / Normal / Bright
Light color
Warm light / RBG lights
Audio level
Low / Medium / Loud
Auto control
Yes / No
SmartHome System
Current System: Amazon Alexa
Digital assistant
Room access selection
Living room, kitchen, dining room
Mode
Standby / Active
Audio level
Low / Medium / Loud


SmartHome System
Current Room: Master Bathroom
Smart Bathroom
Mode
Shower / Bath
Temperature
Cold / Warm / Hot
Auto control
Yes / No
SmartHome System
Current Room: Master Bedroom
Heating and Cooling
Return to home
Set temp preference
73°F
Set time period
10 AM - 10 PM


SmartHome System
Current Status: Auto on
Audio visual system
Light brightness preference
Dim to medium bright
Light color preference
Warm light
Audio level preference
Medium low to Medium
Set time period
5 PM - 9:30 PM
SmartHome System
Current Status: Auto on
Smart Bathroom
Set time period
10 PM - 11 PM
Temperature preference
Hot


SmartHome System
Smart Appliance Control
Refrigerator
More Settings
Microwave
More Settings
Toaster
More Settings
Oven
More Settings
Washer/Dryer
More Settings
Dishwasher
More Settings
Coffee Maker
More Settings
Return to home
SmartHome System
Voice Control via Amazon Alexa
Push Mic to Activate Voice Control
Make commands
Return to home


SmartHome System
Leak Detection
Pipe A: No Leak
Shut Off
Pipe B: No Leak
Shut Off
Pipe C: No Leak
Shut Off
Pipe D: Active Leak
Shut Off
Return to home


SmartHome System
Security Status
No updates available. Home is secure.
Enable SMS Status Report Notifications
Return to home

SmartHome System

Access Control

Time	In/Out	Where
November 27		
8:00AM	Out	Front
8:30AM	Out	Garage
2:15PM	In	Front
6:30PM	Out	Back
9:30PM	In	Front
November 26		
8:15AM	Out	Garage
5:00PM	In	Garage

SmartHome System

Add/Remove Button

Remote Controller

Surveillance

Heating and Cooling

Garage

SmartHome System

\+ Done

Remote Controller

Surveillance -

Heating and Cooling -

Garage -

SmartHome System

Add Timer

Live-in-Lock

Air Conditioning Edit

Lights Edit

TV Edit

SmartHome System

Add Lock

Lock Controller

Front

Back

Garage

Algorithm Design (Pseudo-code):

```
public class Company{
	private String typeOfCompany;
	private User user;
	public Company(User s, String t){...}

	public String getName(){
		return name;
	}
	public String getAddress(){
		return address;
	}
}

public class User {
	private String username;
	private String password;
	private String email;
	private String firstName;
	private String lastName;
	private int ID;

	public User() { //constructor
	… }

	public String getUsername() {
		return username;
	}
	public void setUsername(String username){
		this.username = username;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password){
		this.password = password;
	}
	public String getEmail() {
		return email;
	}
	public void setEmail(String email){
		this.email = email;
```

```
	}
	public String getFistName() {
		return firstName;
	}
	public void setFirstName(String firstName){
		this.firstName = firstName;
	}
	public String getLastName() {
		return firstName;
	}
	public void setLastName(String lastName){
		this.lastName = lastName;
	}
}

public class Account{
	private List<Logs> logs = new ArrayList<Logs>();
	private eventDescription eventDesc = new eventDescription();
	private ArrayList<String> events = new ArrayList<String>();
	private String systemName;
	boolean isArmed;

	public Account()
	{ //constructor
		… }
	public static void addEvent(){
		logs.addEvent(new Logs());
	}
	public List<String> getEvents(){
		return events;
	}
	public static void clearLogs(){
		logs.clearLogs(new Logs());
	}
	public String getEventDescription(){
		return eventDescription;
	}
	public getEventDate(){
		return eventDate;
	}
	public String getName(){
		return name;
	}
	public boolean isArmed(){
```

```
                return isArmed;
        }
        public static void setArmed(boolean armed){
                isArmed = armed;
        }

}

public class System{
        private String systemName;
        private boolean isArmed;
        public System(){ //Constructor
                … }

        public String getName(){
                return systemName;
        }
        public boolean isArmed(){
                return isArmed;
        }
        public void setArmed(isArmed){
                this.isArmed = isArmed;
        }
}

public class Logs{
        private String eventDescription;
        private int eventDate;
        private List<String> events;
        public Logs(){ //Constructor
                … }

        public void addEvent(){
                events.addEvent(new Logs());
        }
        public List<String> getEvents(){
                return events;
        }
        public void clearLogs(){
                events.clear();
        }
        public String getEventDescription(){
                return eventDescription;
        }
```

```
        public int getEventDate(){
                return eventDate();
        }
}


public class SystemControls{
        private boolean isLightOn;
        private boolean isTempOn;
        private int tempPreference;
        private boolean isEntranceSensorActivated;
        private String remoteButtonSelected;
        private int liveInLockTimer;
        private boolean isStatusReportOn;
        private String notificationUpdates;

        public SystemControls(){ //Constructor
                … }

        public void adjustTemp(boolean adjustTemp){
                isTempOn = adjustTemp;
        }
        public void adjustLight(boolean adjustLight){
                isLightOn = adjustLight;
        }
        public int getTempPreference(){
                return tempPreference;
        }
        public void setTempPreference(boolean tempPref){
                tempPreference = tempPref;
        }
        public int autoTempControl(){
                //implementation
        }
        public void setIsEntranceSensorActivated(boolean entranceSenActivate){
                isEntranceSensorActivated = entranceSenActivate;
        }
        public boolean getIsEntranceSensorActivated(){
                return isEntranceSensorActivated;
        }
        public void setRemoteButtonSelected(String remoteButtonSelect){
                remoteButtonSelected = remoteButtonSelect;
        }
```

```
        public String getRemoteButtonSelected(){
                return remoteButtonSelected;
        }
        public void setLiveInLockTimer(int lockTimer){
                liveInLockTimer = lockTimer;
        }
        public int getLiveInLockTimer(){
                return liveInLockTimer;
        }
        public void setNotificationUpdates(String notifUpdate){
                notificationUpdates = notifUpdate;
        }
        public String getNotificationUpdates(){
                return notificationUpdates;
        }
}

public class GeofencingSystemControls{
        private SystemControls geofenceSystem;
        private boolean isFenceOn;

        public GeofencingSystemControls(){ //Constructor
                … }
        public boolean getFenceControl(){
                return isFenceOn;
        }
        public void setFenceControl(boolean fenceControl){
                isFenceOn = fenceControl;
        }


}

public class PetMonitorSystemControls{
        private SystemControls petMonitorSystem;
        private boolean isCameraOn;

        public PetMonitorSystemControls(){ //Constructor
                … }
        public void adjustCamera(boolean adjustCamera){
                isCameraOn = adjustCamera;
        }
        public void alertUnusualBehavoir(){
                //implementation
```

```
        }

}

public class WindowBlindSystemControls{
        private SystemControls windowBlindSystem;
        private boolean isBlindOpen;
        private int adjustBlindWidth;

        public WindowBlindSystemControls(){ //Constructor
                … }
        public void adjustBlind(boolean openClose){
                isBlindOn = openClose;
        }
        public void manualBlind(int manualPref)
        {
                IF (manualPref <= 100 && manualPref <= 0)
                        adjustBlindWidth = manualPref;
        }
}

public class AutoGarageDoorSystemControls{
        private SystemControls autoGarageDoorSystem;
        private boolean isGarageOpen;
        private int[] lockPreferences;

        public AutoGarageDoorSystemControls(){ //Constructor
                … }
        public void adjustGarage(boolean openClose){
                isGarageOpen = openClose;
        }
        public void setLockPreferences(int[] lockPrefs){
                lockPreferences = lockPrefs;
        }
        public int[] getLockPreferences(){
                return lockPreferences;
        }
}

public class YardSystemControls{
        private SystemControls yardSystem;
        private boolean isYardLightOn;
        private boolean isSprinklerOn;
```

```
        public YardSystemControls(){ //Constructor
                … }
        public int adjustYardLight(boolean onOff){
                isYardLightOn = onOff;
        }
        public int adjustSprinkler(boolean onOff){
                isSprinklerOn = onOff;
        }
}

public class AudioAndVisualSystemControls{
        private SystemControls audioVisualSystem;
        private String audioVisualPreference;
        private boolean isVoiceControlOn;

        public AudioAndVisualSystemControls(){ //Constructor
                … }
        public String getAudioVisualPreference(){
                return audioVisualPreference;
        }
        public void setAudioVisualPreference(String audioVisPref){
                audioVisualPreference = audioVisPref;
        }
        public String autoAudioVisualControl(){
                //implementation
        }
        public void setIsVoiceIntegrationActive(boolean voiceContOn){
                isVoiceControlOn = voiceContOn;
        }
        public boolean getIsVoiceIntegrationActive(){
                return isVoiceControlOn;
        }

}

public class DigitalAssistantSystemControls{
        private SystemControls digitalAssistantSystem;
        private String[] digitalAssistantPreference;

        public DigitalAssistantSystemControls(){ //Constructor
                … }
        public String[] getDigitalAssitancePreference(){
                return digitalAssistantPreference;
        }
```

```
        public void setDigitalAssitancePreference(String[] digAssistantPref){
                digitalAssistantPreference = digAssistantPref;
        }
        public String[] autoDigitalAssistanceControl(){
                //implementation
        }
}

public class SmartBathroomSystemControls{
        private SystemControls smartBathroomSystem;
        private String bathroomPreference;
        private boolean isSmartApplianceOn;
        private boolean isLeakSensorOn;

        public SmartBathroomSystemControls(){ //Constructor
                … }
        public String getBathroomPreference(){
                return bathroomPreference;
        }
        public void setBathroomPreference(String bathroomPref){
                bathroomPreference = bathroomPref;
        }
        public String autoBathroomControl(){
                //implementation
        }
        public void setSmartAppliance(boolean smartApplianceSet){
                isSmartApplianceOn = smartApplianceSet;
        }
        public boolean getSmartAppliance(){
                return isSmartApplianceOn;
        }
        public void setLeakDetection(boolean leakDetection){
                isLeakSensorOn = leakDetection;
        }
        public boolean getLeakDetection(){
                return isLeakSensorOn;
        }
}


public class AccessControl{
        private SystemControls accessControlSystem;
        private bool isEntranceSensorActivated;
```

```
	public accessControlControls(){ //Constructor
		… }
	public bool getIsEntranceSensorActivated(){
		return isEntranceSensorActivated;
	}
	public void setEntranceActivator(bool entranceSensor){
		isEntranceSensorActivated = entranceSensor;
	}
	public void handleSensor(bool sensor) {
		//if the sensor is activated
			//get the time, if it was in/out, and which entrance
			//print these variables on the log
		//else
			//nothing, wait for when the sensor is activated
	}
}
```

```
public class RemoteController{
	private SystemControls remoteControlSystem;
	private String[] currentButtons;
	private String remoteButtonSelected;

	public remoteControlControls(){ //Constructor
		… }
	public bool getRemoteButtonSelected(){
		return remoteButtonSelected;
	}
	public void setRemoteButtonSelected(String button){
		remoteButtonSelected= button;
	}
	public void handleButtonSelected(String button) {
		//iterate through the array of buttons until a name matches the parameter
		//if there's a match
			//shortcut to the system page
	}
	public int addButton(String button) {
		//add String to array of current buttons
		//return 1 if successful
	}
	public int addButton(String button) {
		//iterate through current Buttons for a matching String
			//if there's a match
				//remove String from array of current buttons
		//return 1 if successful
	}
}

public class LockControl{
	private SystemControls lockControlSystem;
	//lock preferences where 1 means locked and 0 means unlocked
	private int[] lockPreferences;

	public lockControlControls(){ //Constructor
		… }
	public int[] getLockPreferences(){
		return lockPreferences;
	}
	public void setLockPreferences([int[] lockpref) {
		lockPreferences= lock;
	}
}
```

```
public class LiveInLock{
	private SystemControls liveInLockSystem;
	private String systemName;
	private int timer;

	public liveInLockControls(){ //Constructor
		… }
	public String getSystemName(){
		return systemName;
	}
	public void setSystemName(String name){
		systemName= name;
	}
	public int getTimer(){
		return timer;
	}
	public void setTimer(int time){
		timer= time;
	}
	public void handleLiveInLock(String name, int time) {
		//access the system associated with the name
		//if the time is equal to the timer plus the time when the timer was made
			//adjust the controls for that system
	{
}

public class WaterLeakDetection {
	Private SystemControls waterLeakSystem;
	Private boolean isLeakSensorOn;

public WaterLeakDetection() {
//constructor
//initialization
}

public void setWaterLeakDetection(boolean waterLeakDetection) {
	isLeakSensorOn = waterLeakDetection;
}

public boolean getWaterLeakDetection() {
	Return isLeakSensorOn;
}
```

```
public void handleWaterLeak() {
        if (isLeakSensorOn) {
                //trigger alarm when leak detected
                //shut off water supply
                //log event in system logs
                System.out.println("Leak detected. Shutting off water");
                waterLeakSystem.adjust(false);
}
}
}

public class SmartApplianceControls {
        Private SystemControls smartApplianceSystem;
        Private boolean isSmartApplianceOn;

public SmartApplianceSystemControls() {
        //constructor
        //Initialize
}
public void setSmartAppliance(boolean smartApplianceSet) {
        isSmartApplianceOn = smartApplianceSet;
}

public boolean getSmartAppliance() {
        Return isSmartApplianceOn;
}
public void controlSmartAppliance() {
        if(isSmartApplianceOn) {
                //adjust settings
}
}
}

public class StatusReportSystemControls {
        Private SystemControls statusSystem;
        Private boolean isStatusReportOn;

public StatusReportSystemControls() {
        //constructor + initializations
}
Public void setStatusReport(boolean statusReportSet) {
        isStatusReportOn = statusReportSet;
}
public boolean getStatusReport() {
```

```
        Return isStatusReportOn;
}
public String generateStatusReport() {
        if(isStatusReportOn) {
                //collect info from status report
                String systemName = securitySystemStatus.getName();
                boolean isActive = securitySystemStatus.isActive();
                //all other security measures

                String statusReport = “Security Status Report: …“ ;

                return statusReport; }
else { return “Status report not enabled”;
}
}
}


public class VoiceControlSystemControls {
        private SystemControls voiceControl;
        private boolean isVoiceControlOn;

public VoiceControlSystemControls() {
        //with constructor and initialization
}
public void setIsVoiceControlActive(boolean voiceControlSet) {
        isVoiceControlOn = voiceControlSet;
}
public boolean getIsVoiceControlActive() {
        return isVoiceControlOn;
}
public void activateVoiceControl(0 {
        if(isVoiceControlOn) {
                //logic to press microphone button to activate voice control
                System.out.println(“Voice Control Enabled. Make Commands.”);
}
}
//same thing for deactivating voice control

public void processVoiceCommand(String voiceCommand) {
If (isVoiceControlOn) {
//audio queues sent as strings to Amazon Alexa digital assistant and processed
}
}
```

```
}
```