```
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
from kivy.uix.button import Button
from kivy.uix.label import Label
from kivy.uix.textinput import TextInput

class SmartHomeSecuritySystem(BoxLayout):
    def __init__(self, **kwargs):
        super(SmartHomeSecuritySystem, self).__init__(**kwargs)
        self.orientation = 'vertical'


        self.lock_status = Label(text="Lock: Locked")
        self.add_widget(self.lock_status)

        self.door_status = Label(text="Door/Window: Closed")
        self.add_widget(self.door_status)

        self.motion_status = Label(text="Motion: Cleared")
        self.add_widget(self.motion_status)

        self.camera_status = Label(text="Camera: Off")
        self.add_widget(self.camera_status)

        self.alarm_status = Label(text="Alarm: Off")
        self.add_widget(self.alarm_status)


        self.log_display = TextInput(size_hint_y=None, height=200, readonly=True)
        self.add_widget(self.log_display)


        control_buttons = BoxLayout(size_hint_y=None, height=50)
        self.add_widget(control_buttons)

        lock_button = Button(text="Lock Door", font_size=13)
        lock_button.bind(on_press=self.lock_door)
        control_buttons.add_widget(lock_button)

        unlock_button = Button(text="Unlock Door", font_size=13)
        unlock_button.bind(on_press=self.unlock_door)
        control_buttons.add_widget(unlock_button)

        open_button = Button(text="Open Door/Window", font_size=13)
        open_button.bind(on_press=self.open_door)
        control_buttons.add_widget(open_button)

        close_button = Button(text="Close Door/Window", font_size=13)
```

```
        close_button.bind(on_press=self.close_door)
        control_buttons.add_widget(close_button)

        detect_button = Button(text="Detect Motion", font_size=13)
        detect_button.bind(on_press=self.detect_motion)
        control_buttons.add_widget(detect_button)

        clear_button = Button(text="Clear Motion", font_size=13)
        clear_button.bind(on_press=self.clear_motion)
        control_buttons.add_widget(clear_button)

        record_button = Button(text="Start Recording", font_size=13)
        record_button.bind(on_press=self.start_recording)
        control_buttons.add_widget(record_button)

        stop_record_button = Button(text="Stop Recording", font_size=13)
        stop_record_button.bind(on_press=self.stop_recording)
        control_buttons.add_widget(stop_record_button)

        toggle_alarm_button = Button(text="Toggle Alarm", font_size=13)
        toggle_alarm_button.bind(on_press=self.toggle_alarm)
        control_buttons.add_widget(toggle_alarm_button)

    def lock_door(self, instance):
        self.lock_status.text = "Lock: Locked"
        self.log_action("Door locked.")

    def unlock_door(self, instance):
        self.lock_status.text = "Lock: Unlocked"
        self.log_action("Door unlocked.")

    def open_door(self, instance):
        self.door_status.text = "Door/Window: Opened"
        self.trigger_alarm()
        self.log_action("Door/Window opened.")

    def close_door(self, instance):
        self.door_status.text = "Door/Window: Closed"
        self.log_action("Door/Window closed.")

    def detect_motion(self, instance):
        self.motion_status.text = "Motion: Detected"
        self.trigger_alarm()
        self.log_action("Motion detected.")

    def clear_motion(self, instance):
        self.motion_status.text = "Motion: Cleared"
        self.log_action("Motion cleared.")
```

```
    def start_recording(self, instance):
        self.camera_status.text = "Camera: Recording"
        self.log_action("Camera started recording.")

    def stop_recording(self, instance):
        self.camera_status.text = "Camera: Off"
        self.log_action("Camera stopped recording.")

    def toggle_alarm(self, instance):
        current_status = self.alarm_status.text.split(": ")[1]
        new_status = "On" if current_status == "Off" else "Off"
        self.alarm_status.text = f"Alarm: {new_status}"
        self.log_action(f"Alarm toggled to {new_status}.")

    def trigger_alarm(self):
        print("WARNING! Security breach detected!!!")
        self.log_action("WARNING! Security breach detected!!!")

    def log_action(self, action):
        self.log_display.text += f"{action}\n"
        self.log_display.cursor = (0, 0)  # Scroll to the top

class MyApp(App):
    def build(self):
        return SmartHomeSecuritySystem()

if __name__ == '__main__':
    MyApp().run()
```