Develop a Smart Home Automation System The system should allow users to remotely control and monitor home devices like lights, fans, air conditioning, security systems, and more.

Users should be allowed to turn devices ON/OFF, adjust settings (e.g., temperature, brightness), and schedule automation. It should implement user authentication and role-based access (Admin vs. Regular User). Features like Real-time monitoring of device status, energy consumption should be included. It should support Event-driven automation, such as turning on lights when motion is detected. Ensure secure communication and data handling. Implement a user-friendly GUI or command-line interface.

These basic minimum features are essential, but to establish a comprehensive system, additional functionalities may need to be incorporated.

The entire system has to be designed around OOP using Java.

Minimum 6 classes are required to accommodate all the requirements specified in the design problem. Additionally, it should include the following:

1. Overloaded methods (minimum 2)
2. Overloaded constructors (minimum 2)
3. Vararg overloading (minimum 2)
4. Nested classes (static or nonstatic, atleast 1, this is a part of I above)
5. Abstract class (minimum 1)
6. Interface (minimum 1, it can be nested interface or single level or multiple inheritance)
7. Hierarchical Inheritance (atleast 1)
8. Multiple Inheritance (atleast 1, this should be in addition to VI above) (IX)
9. Wrappers
10. Package
11. Exception handling (atleast two cases)
12. I/O: File Handling, scanner class etc. (atleast one from each of these)
13. Multithreading (by either Implementing the Runnable interface or extending the thread class)

These are the minimum requirements, but you have the freedom to incorporate a greater number of each as needed.

Note: The names of variables, methods and classes should be lexically rational and should be accompanied with their description in the comments alongside. Your overall code will be something similar to the codes that you wrote in most of your lab hours. It goes without saying that the code should be well indented and should compile and run error free. However, do remember that a non-running code is better than a plagiarized one, and hence the latter one will be penalised heavily if it exceeds 10-15% (may result in recession of this component).