Report on gRPC Project

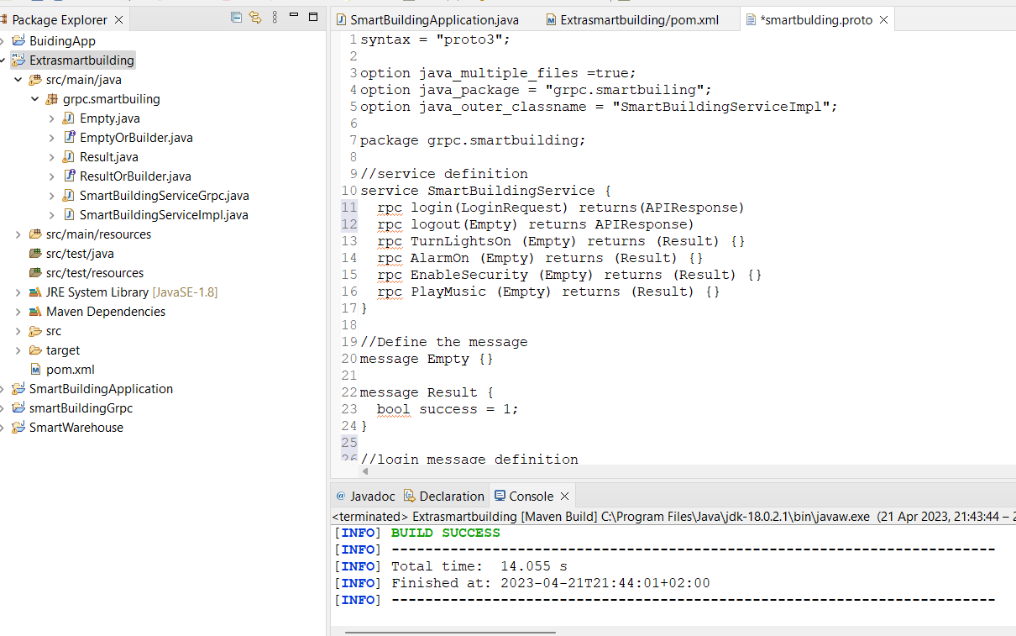
 Service Definitions :proto3 file

Login and Logout gRPC services were defined in one proto file

The file is shown below. The smart warehouse I created has a login functionality, so I chose to implement this as one of my services.

The LoginRequest takes in 2 parameters username and password and returns an APIResponse in the form of a message.

The Logout has no parameters defined and returns an empty string.

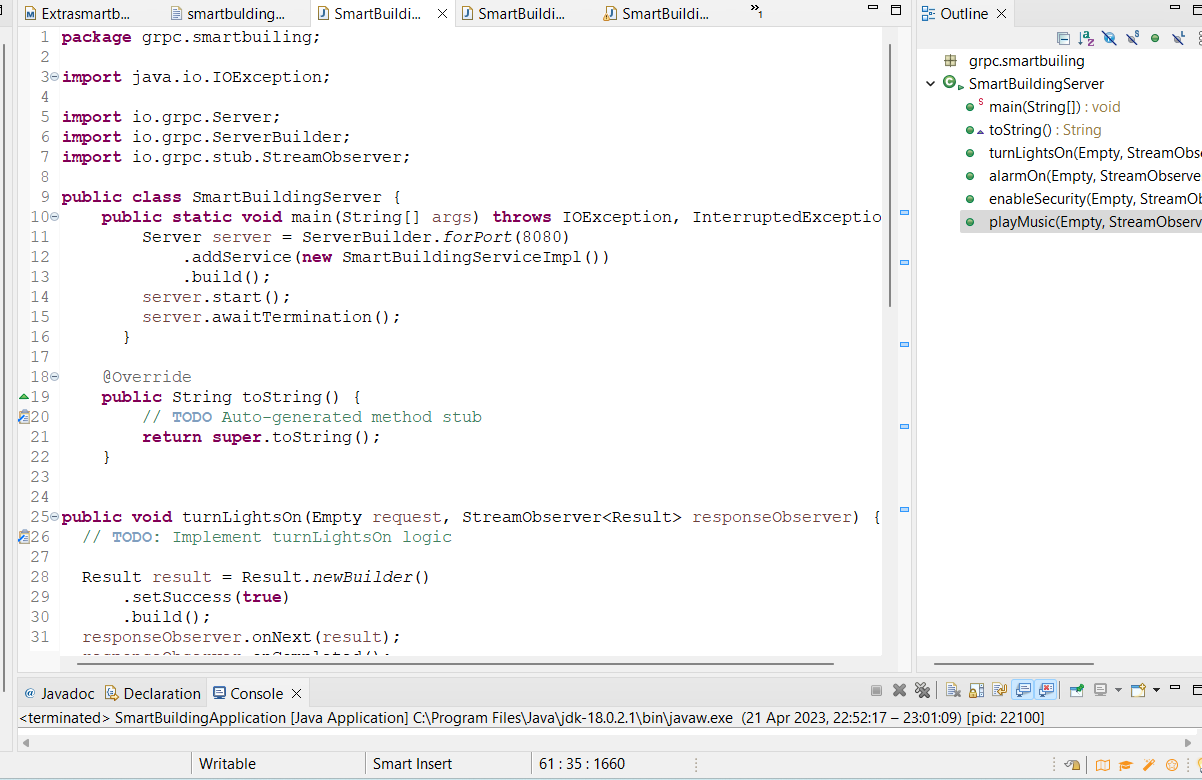


 Service Implementations : SmartBuildingServer

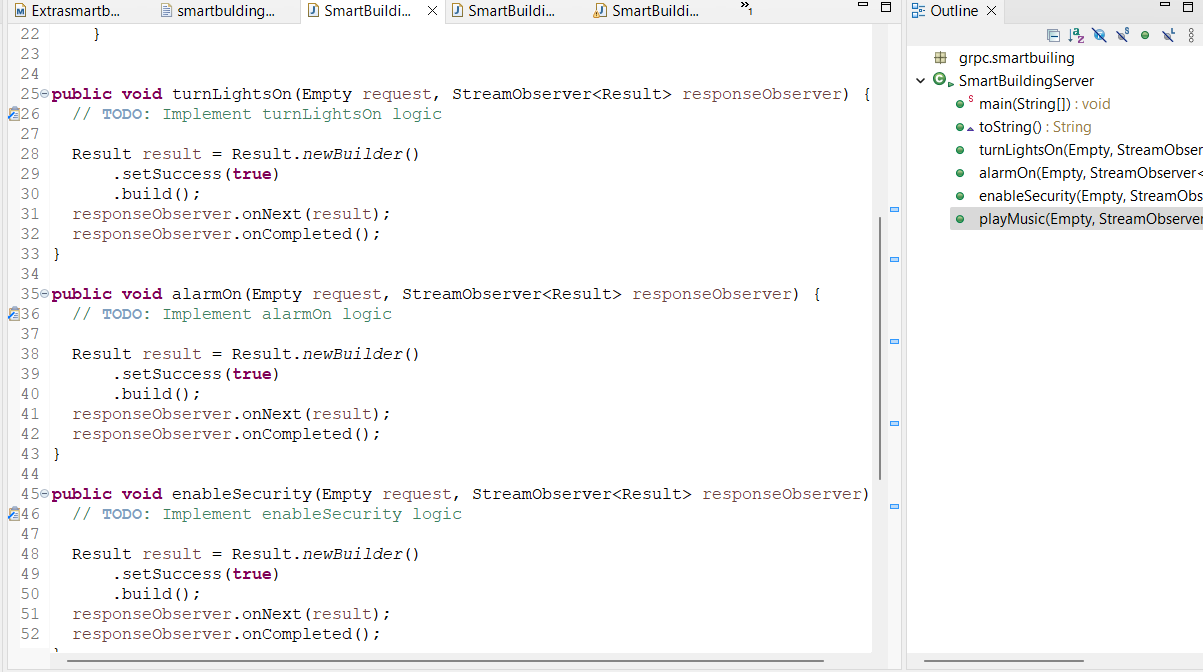
The service implementation is shown in the screenshots below.

The implementation of each method depended on the specific logic for each service. The GUI application returns a true when the service has been activated,so I implemented each service to return a result which I defined in the proto file. After the logic implementation, I created a response message using the protobuf builder classes and send it back to the client using the responseObserver

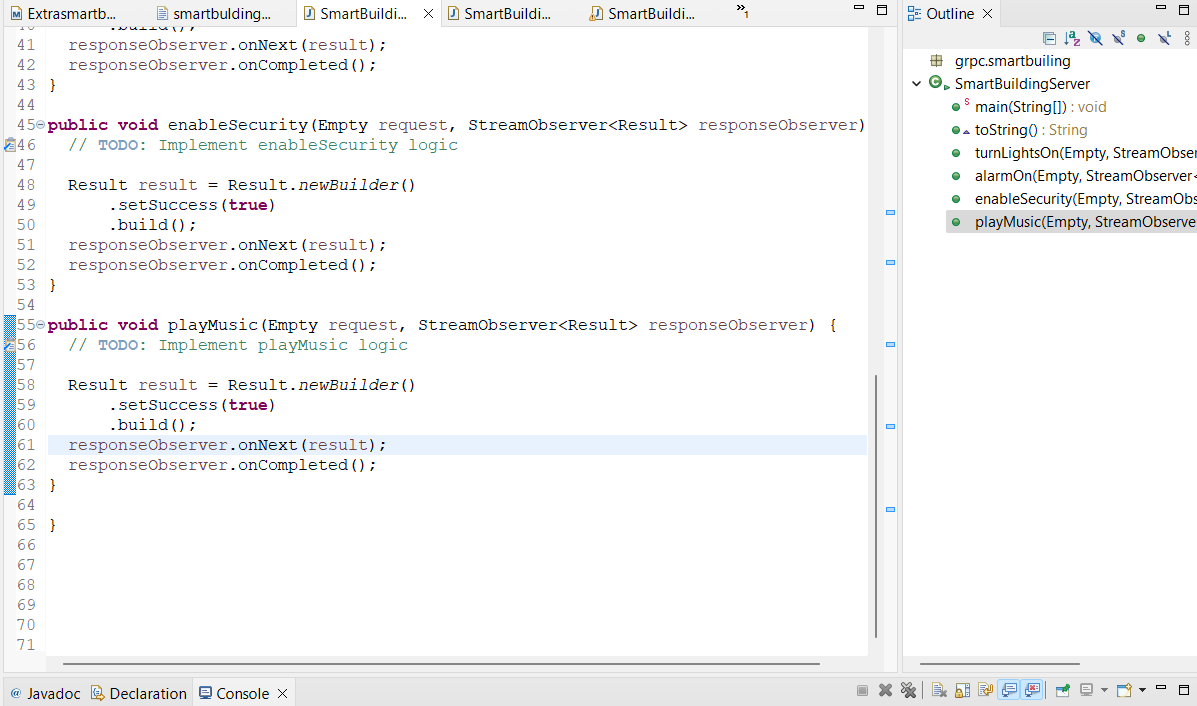
The code below, shows the main method and the server code.



The code below shows the turnLightsOn method, alarmOn and security enabled methods.



The code below shows the security enabled and playmusic methods.



 Naming Services.

//service definition

service SmartBuildingService {

rpc login(LoginRequest) returns(APIResponse)

rpc logout(Empty) returns APIResponse)

rpc TurnLightsOn (Empty) returns (Result) {}

rpc AlarmOn (Empty) returns (Result) {}

rpc EnableSecurity (Empty) returns (Result) {}

rpc PlayMusic (Empty) returns (Result) {}

}

 Remote Error Handling & Advanced Features

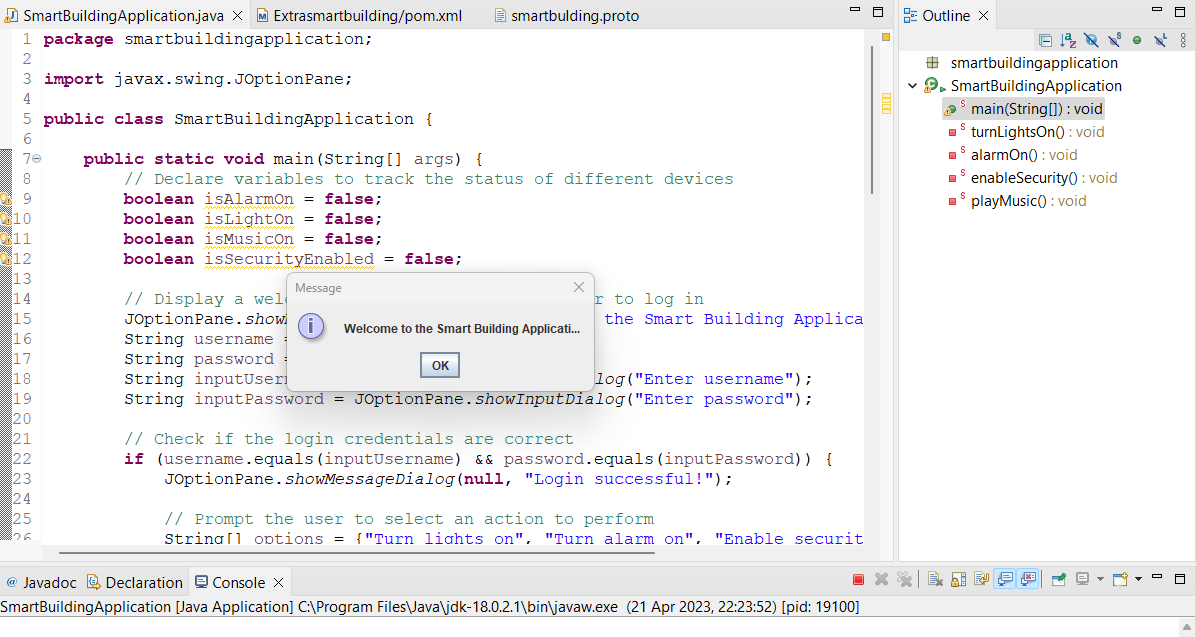
 Client - Graphical User Interface (GUI)

The Client Service is a simple GUI implemented using JOption Pane.

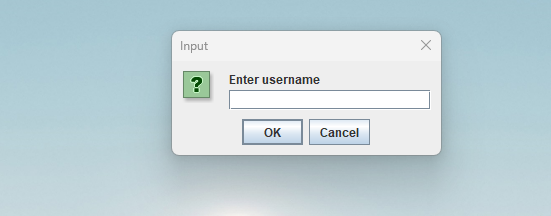
On running the application, the first interface is a Welcome message to the user, who can then click Ok To proceed with using the system.

The login options are hard coded and the login details are as follows

username = "user"; password = "1234";

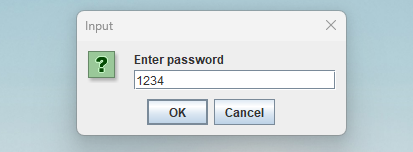


The username interface is as follows

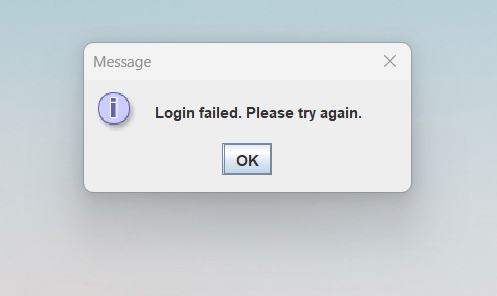
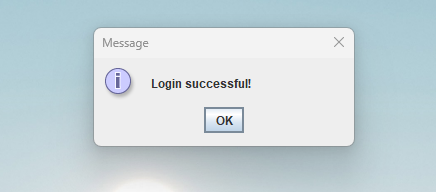


The user then enters the username :user and clicks ok to login or cancel to stop the process

If the user proceeds to enter their username, the following Window is displayed, prompting the user to enter the password.



The user then gets confirmation of Login Success if they have provided the correct credentials, if not then they get a Login Unsuccessful which cancels the whole process



The next Window gives the user the access to the smart building services which are

Turn lights on

Turn Alarm on

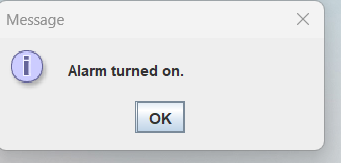
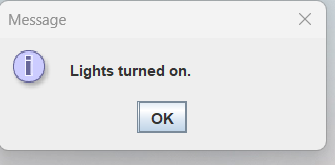
Enable Security

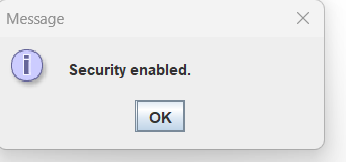
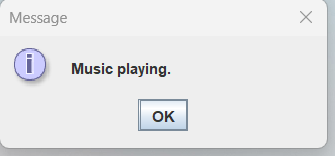
Play Music

There is also an Exit option which is equivalent to login, once one has exited the system, they have to provide their credentials again to log in.



The selected service that has been turned on, returns a confirmation that the service is now enabled, below is the Lights turned on confirmation



 GitHub