CSCI 5448 - Object Oriented Analysis and Design
Project Part 7
#31 Smart Home Management System

Team

1. Raj Kumar Subramaniam
2. Saritha Senguttuvan
3. Savitha Senguttuvan

Vision

Our aim is to develop a Smart Home Network Application software, IoT based application to monitor and configure different sensors in the network.

Project Description

Smart Home Management System is a desktop app, which helps to monitor and update notifications from the smart sensors at a home. The project is assumed to be for a client company, ABC Smart Home Systems which manufactures smart home products, so the Admin will be the company's employee and the customers will be the one who purchases the products from the company. The application will help the customer to configure sensors in the home network and update any notifications to the user. The application will also allow the company employees to add new products into the system which can be used by the customers.

Use Cases Implemented

User Requirement		
User ID	**Description**	**Actors**
UR-01	As an admin/user, I should be able to create a role based account so that I can login	Admin or user
UR-02	As an admin, I should be able to add products into the Product Database	Admin
UR-03	As an admin, I should be able to remove existing products from the Product Database	Admin

UR-08	As a user, I should be able to add sensors to my home network from the Product List	User
UR-09	As a user, I should be able to remove sensors from the Home Sensor Network	User
UR-10	As a user, I should be able to view and configure the settings for each sensor in the Home Sensor Network	User
UR-11	As a user, I should see a page for individual sensor to show sensor value, help option, sensor type	User
UR-13	As a user, I should be able to see the statistics of each sensor in the Home Sensor Network	User
UR-16	As a user, I should be able to create custom group of the sensors in the Home Network so that I can control the group (e.g., Turn off all the sensors in the group)	User
UR-18	As a user, I should be able to set the location of the sensors in the Home Network (eg. In kitchen, in Bedroom)	User
UR-19	As a user, I should be able to set ON/OFF for the group of sensors in the home network	User
UR-20	As a user I should be able to set the threshold value of the sensor in the settings	User
UR-23	As an admin/user, I should be able to login in to the app so that I can access the application	User
UR-25	As an admin, I should be able to view the list of added products	Admin
UR-26	As a user, I should be able to create a home network so that I can manage all the sensors.	User
UR-27	As a user, I should be able to list all the sensors in the network so that I can select to view	User
UR-28	As a user, I should be able to remove groups in the network so that I can control group	User
UR-29	As a user I should be able to add sensors to a group, so that I can control the group	User

UR-30	As a user I should be able to remove sensors from the group, so that I can customize the group	User
UR-31	As a user I should be able to view sensors from the group, so that I can see the information	User
UR-32	As a user I should be able to change the power for the group of sensors, so that group of sensors can be controlled in a single click	User

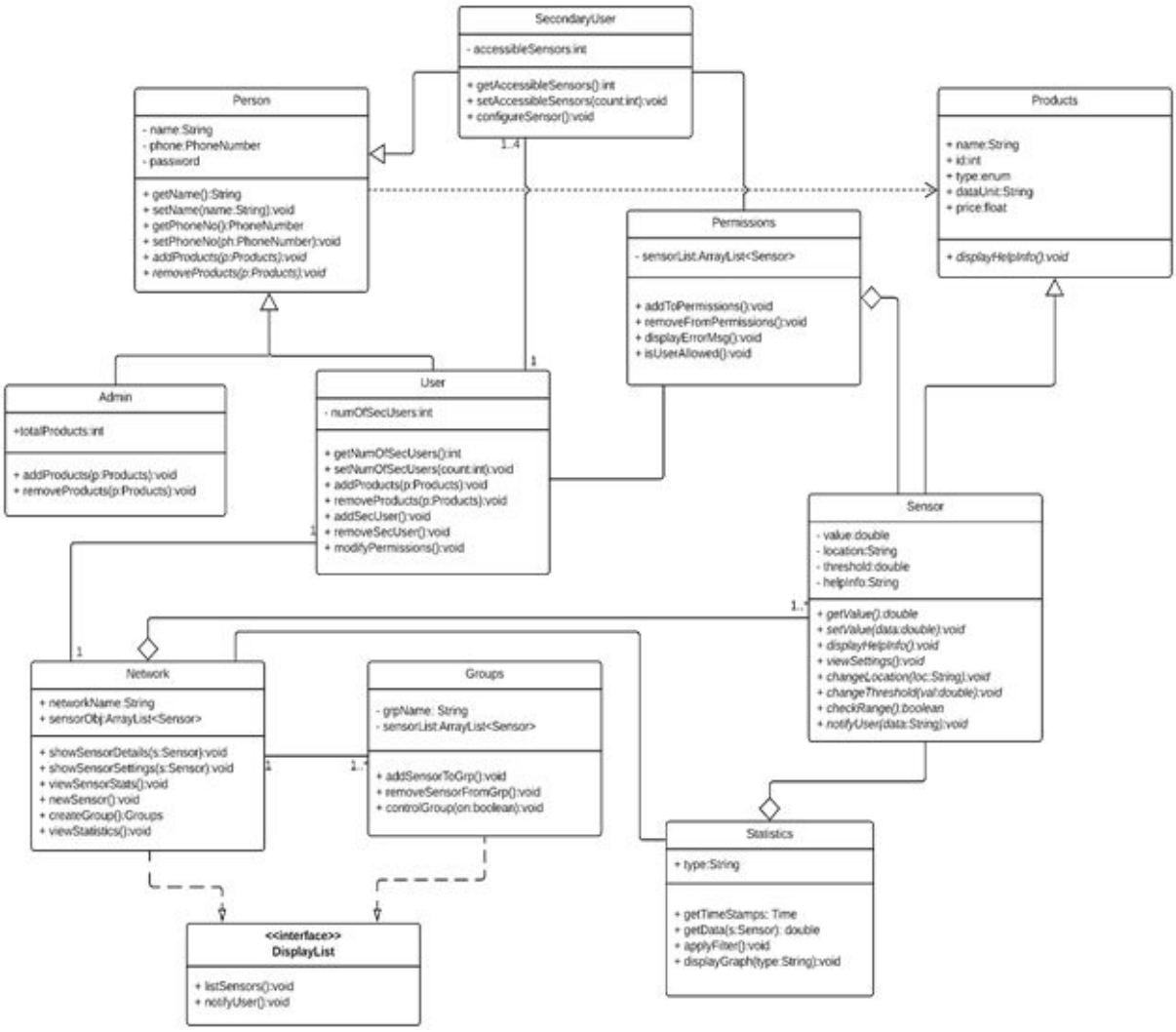
We have also added and implemented more use cases after the initial Requirement analysis.

Use Cases - Dropped

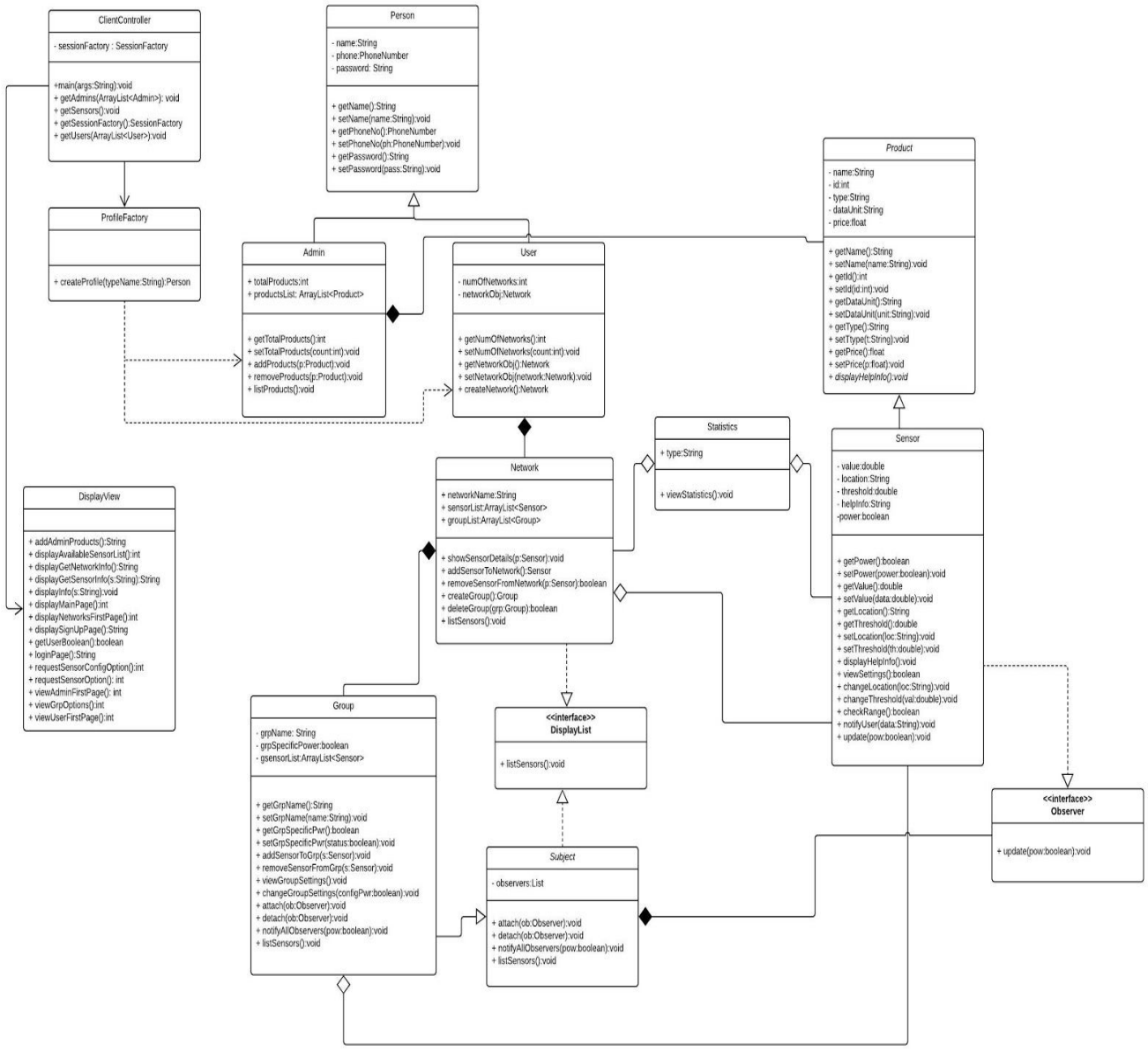
User Requirement		
User ID	**Description**	**Actors**
UR-04	As a user, I should be able to add secondary users to limit their access to sensors in the network (e.g., creating an account for children)	User
UR-05	As a user, I should be able to provide access to a group of sensors to secondary users (other family members)	User
UR-06	As a user, I should be able to edit access to sensors to secondary users	User
UR-07	As a user, I should be able to remove secondary users from the system	User
UR-12	As a user, I should be able to access the help menu to get the details about the sensor	User
UR-14	As a user, I should be able to see the ON/OFF periods of the sensors for the past day	User
UR-15	As a user, I should be able to view the sensor statistics based on certain filters (e.g., during a period, based on sensor type, location etc.)	User
UR-17	As a user, I should get notifications if the sensor data goes beyond the threshold value	User

UR-21	As a user, I should be able to reset the sensor configuration to default settings	User
UR-22	As a user I should be able to raise complaints about a sensor and request for maintenance	User
UR-24	As a user I should be able to reset the password	User

Initial Part-2 Class Diagram



Final Class Diagram



Changes between the first version and last version:

In the progress of the project we made few design changes during the development and also because of the new concepts learned in the class. Doing the design up front helped us to experience the software project flow and made the implementation easier in a structured way.

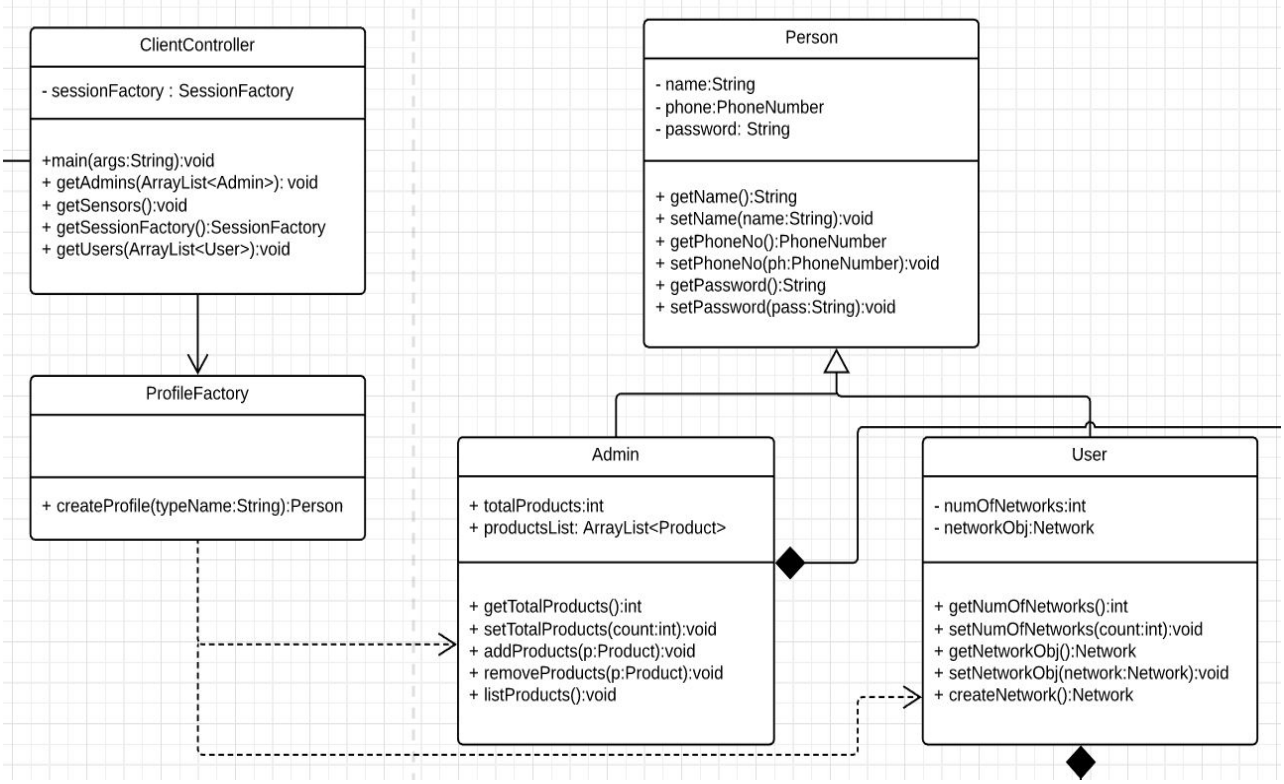
We have added few classes in the final implementation to take advantage of design patterns and also because of few requirement changes. We have implemented, factory design pattern and Observer design pattern to implement a good design of object oriented software.

We have also added DisplayView class to bind the functions related to displaying. This will also help us to make changes in the future if we plan to implement a GUI with graphical interface.

Designing up-front has helped the team to discuss and think a lot before starting the coding. This approach also helped us to split work across the team members and allow developers to work independently without relying on the other team members. The integration was also made easy because of viewing the problem statement based on classes and objects.

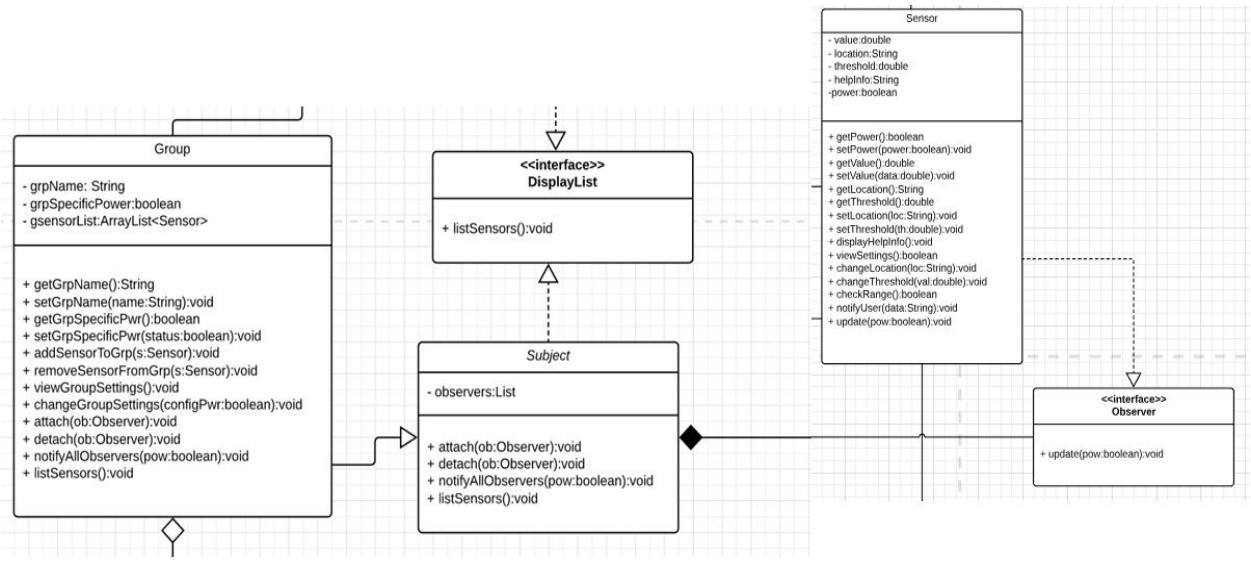
Design Patterns

Factory Design Pattern



We implemented factory design pattern to create objects for Admins and Users based on the user input. This will help us to dynamically create objects for different uses using the same code. The Profile factory class instantiates the classes based on the input from the user and the ClientController class will be able to use the objects with the class name of parent.
Based on the instantiation different operations can be called from the controller and can be used to access options for the Admin and User. This design pattern is also a very useful design in embedded application where the code has to be portable across different hardware platforms.

Observer Design Pattern



We used observer pattern to send notifications to update the sensor settings when the user changes the settings in the group. In the group, multiple sensors will be added and the user has an option to change the settings for all the sensors in the same groups. When the user changes the group settings the Group class will notify all the sensors with help of the Observer interface implemented by the Sensor using the update function. The class diagram above shows our implementation of the Observer design pattern.

Lessons Learned

The process of analysis and design helps a project implementation to go smoother especially for big projects. In big projects, designing helps in separating the work across teams and helps in independent implementation and testing of all modules. It also gives good plan for the integration team to merge all the modules and test the software.

In this project, we have implemented design patterns and Hibernate ORM framework which helped to improve our knowledge on object oriented thinking. Designing a solution based on object oriented approach has given us good hands-on experience and gives an insight into how industries work on big software projects.