Design Documentation

Home Dork

Revision History

Date	Version	Description	Author
2021-09-15	1.0	Modifying Design documentation	Ali Habesh, Amr Al-shaaba, Stiv Abdulwahed, Hani Al-zir.
2021-10-04	1.1	Modified the images, changed POST to PUT, also images were not properly placed.	Ali Habesh, Amr Al-shaaba, Stiv Abdulwahed, Hani Al-zir.
2021-10-22	1.2	UML classes - Api UML classes - Model Use case diagram - Login Use case diagram - general use features	Ali Habesh, Amr Al-shaaba, Stiv Abdulwahed, Hani Al-zir.

Design item List

Requirement Name	Priority
D1. Log in system (API get/post)	Essential
D2. Scrolls & On/Off button (API get/post)	Essential
D3. UML classes - Api	Essential
D4. UML classes - Model	Essential
D5. Use case diagram - Login	Essential
D6. Use case diagram - general use features	Essential

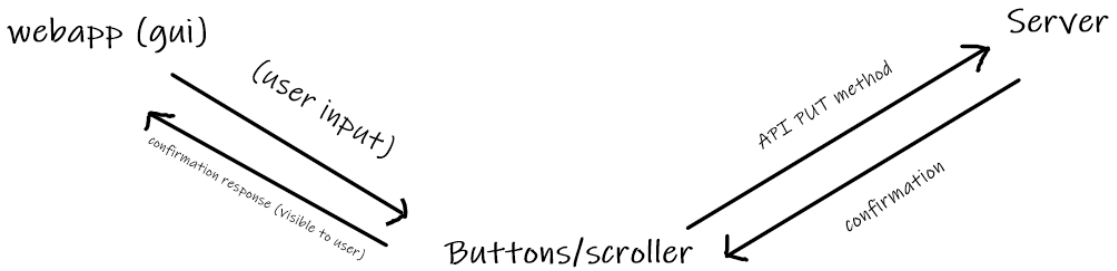
Design Item Descriptions

D1 – Log in

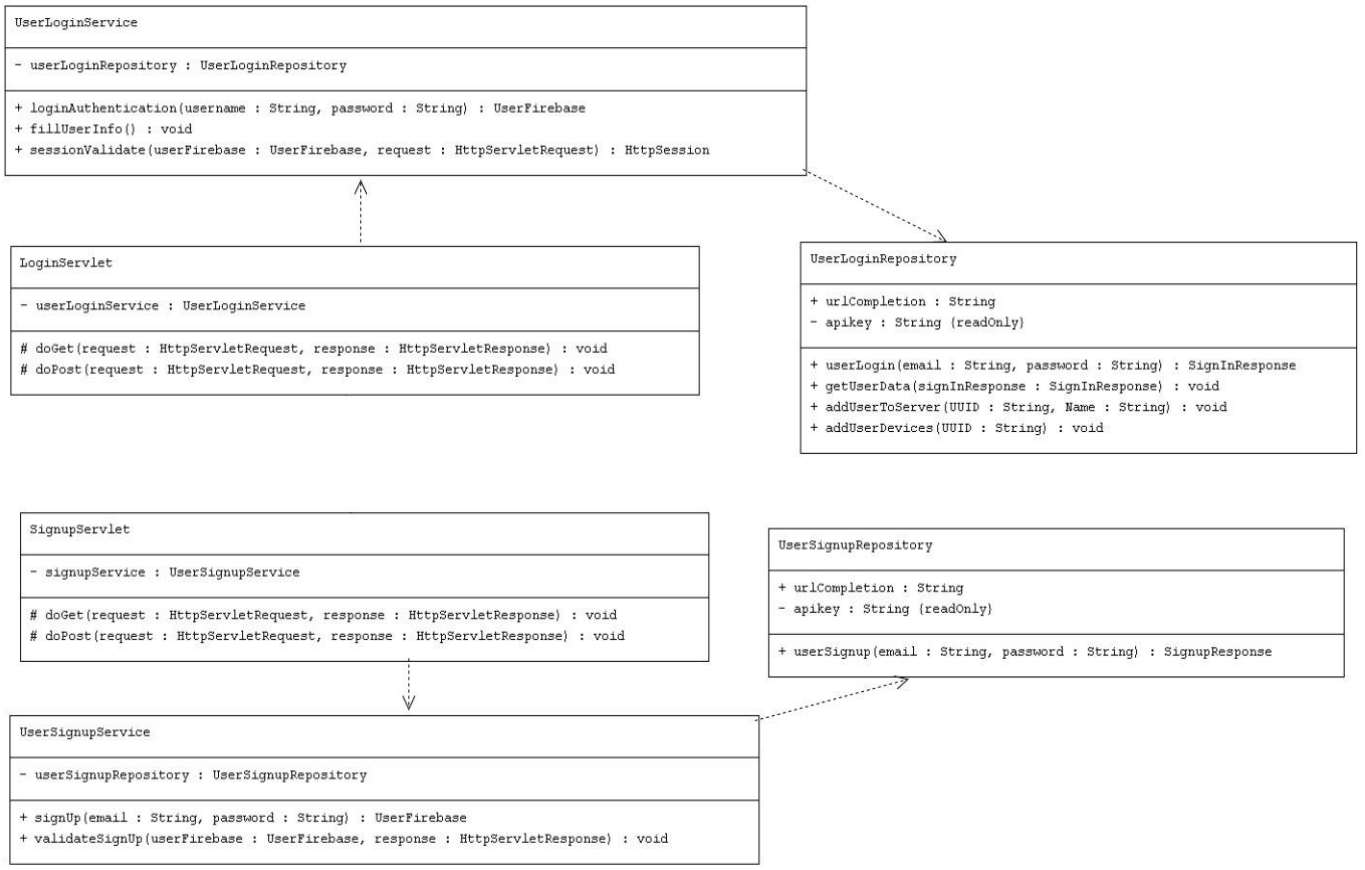
Login system is used for user authentication to be able to access control of their home (smart house). The basic idea is to use the database with some form of security to identity the user before giving him access.

D2 - Scrolls & On/Off button

The user should be able to communicate to his smart house through his private profile using features such as scrollers and buttons. The scroller could be used for a more specific input such as light sensitivity, while the button can be used as more off a binary input (on/off). The buttons will give information and receive information using API's get/post methods, which essentially means the web/phone application will communicate with the server.

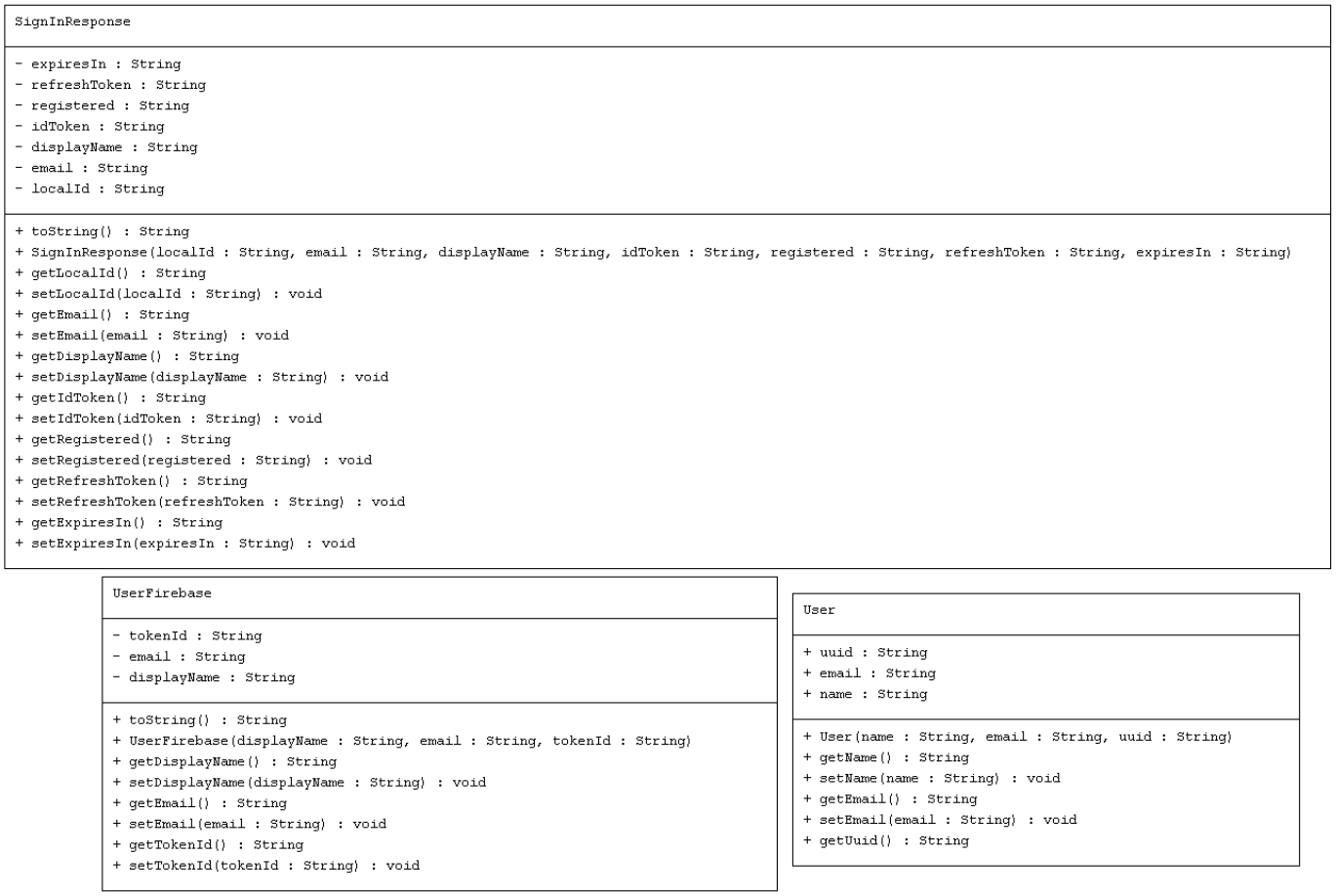


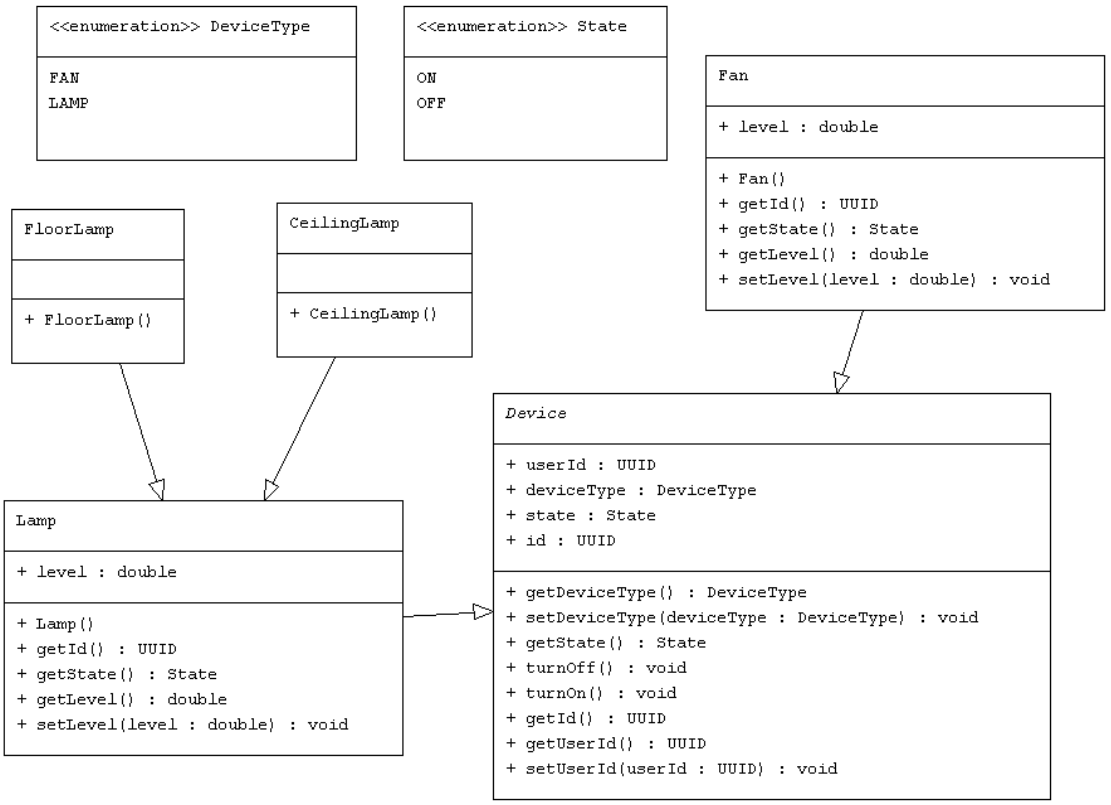
D3. UML class API



- The API UML class diagram shows the underlying structure of the api implementation, these classes are mandatory to create a working environment and to be able to communicate without fail using Homedorks API server.

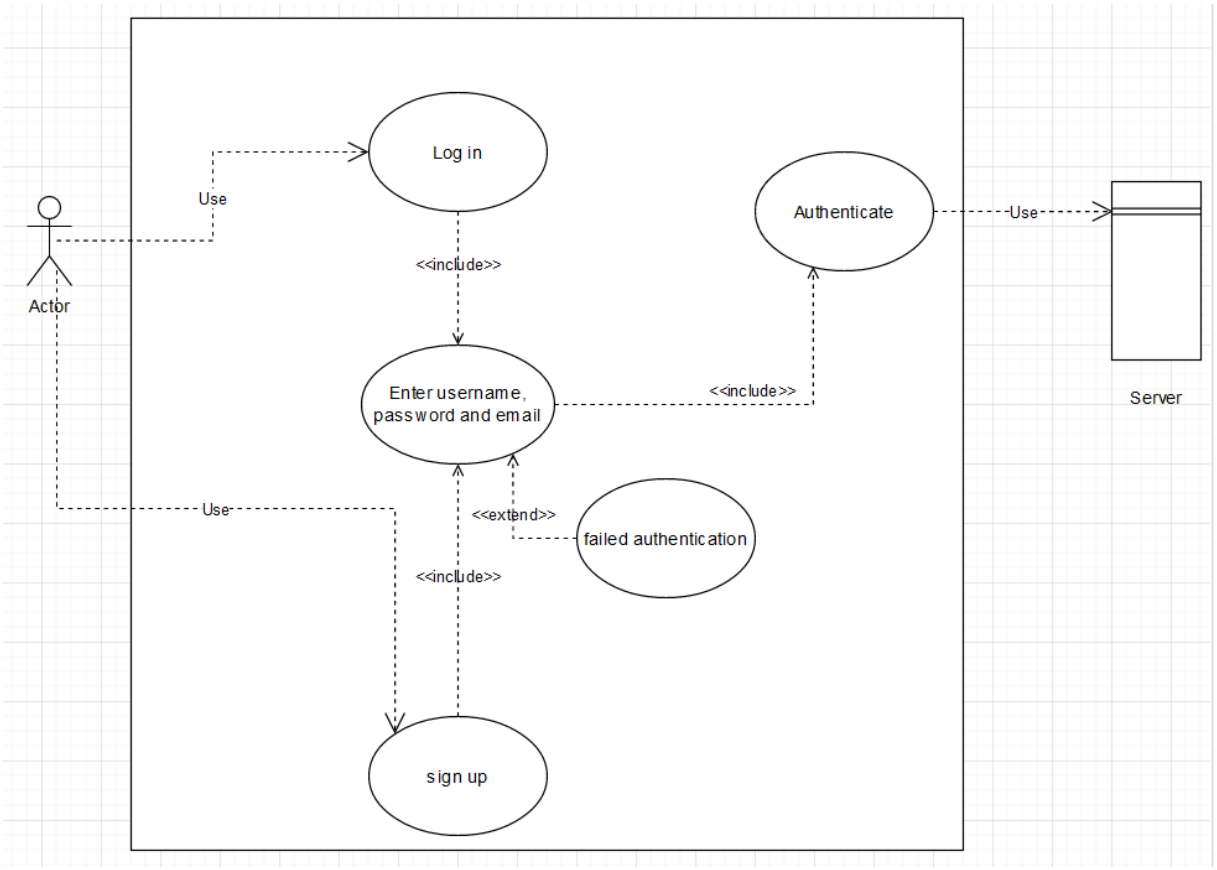
D4. UML classes - Model





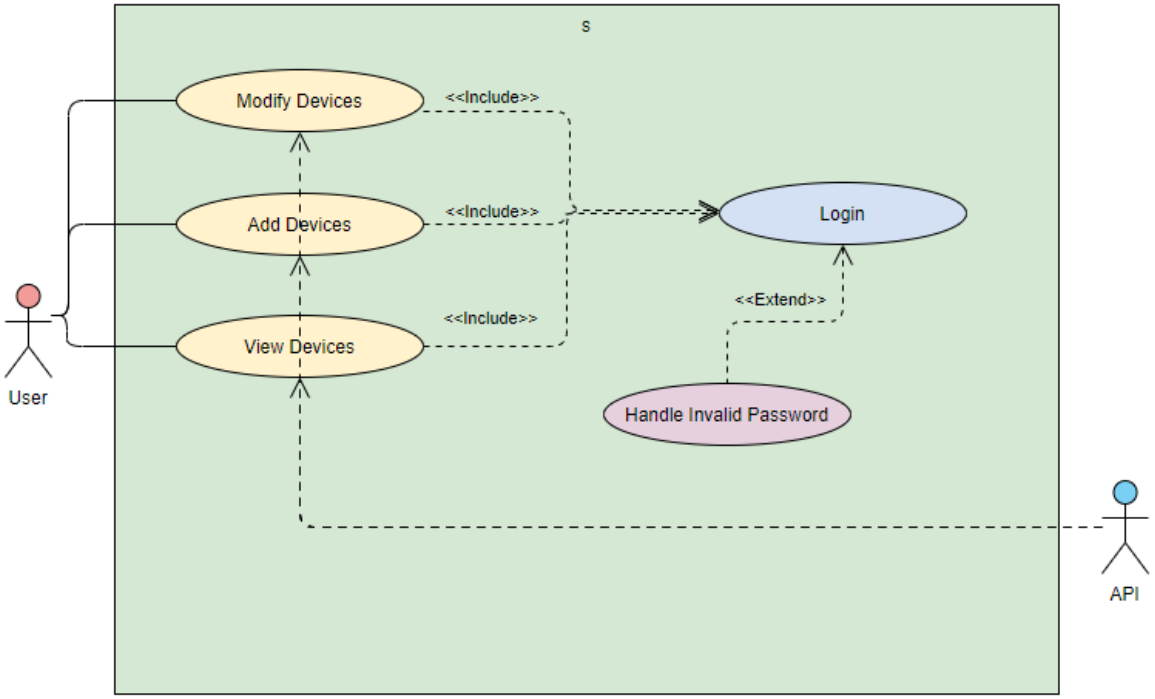
- The Model UML class diagrams depict the required classes in order to consume the Homedork API. Homedork API responds with a set of structured POJO Objects and/or List with POJO objects. In order to receive the response these classes will have to correspond to the server objects.

D5. Use case diagram - Login/Signup



- The login use cases system. The system simply requests the user to input email, password and/or username depending on what you choose, either signup or login. The system then tries to authenticate/check for valid email and either a failed or successful response is expected.

D6. Use case diagram - general use features



- The general use features correspond to the normal API consumption that will take place once the user either turns off/on TV, increases light brightness and much more. This is a general scenario of a functioning Homedork feature system.