Project IOT

|  |
| --- |
| Technical Manual  Software Architecture Manual |

**INTRODUCTION**

The information in this document is provided with the purpose of illustrate the structure of the IOT Project, which includes the process covered to create it, the task model of the project which illustrate the abstract functionality of the system, the activities and process of each activity.

The document includes UML diagrams with software pattern applied, in addition it includes the database modelling and a quick view to API system on flask.

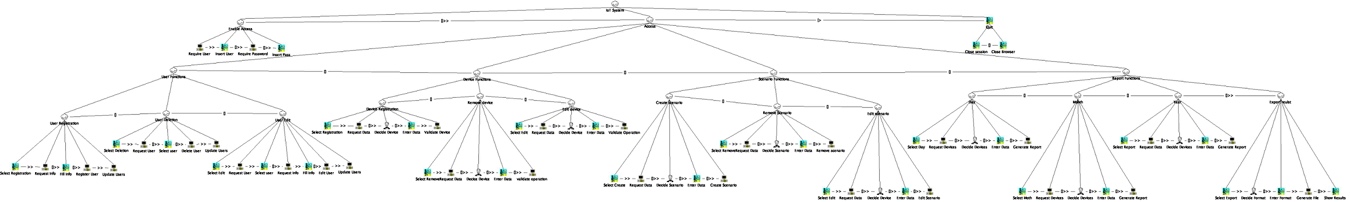
**CONTENTS**

1. **INTRODUCTION**
2. **CONTENTS**
3. **SOFTWARE ARCHITECTURE**
   1. PROCESS COVERED
   2. TASK MODEL
   3. ACTIVITY MODEL
   4. SEQUENCE DIAGRAMS
   5. DABASE MODELING
   6. CLASS DIAGRAM
   7. PATTERNS

**SOFTWARE ARCHITECTURE**

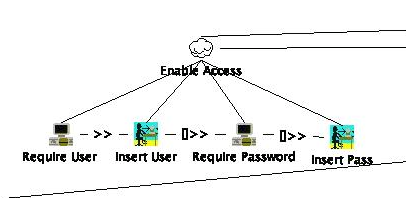
TASK MODEL

The task model provide a description of each task in a workflow of the IOT System, those tasks are Enable Access, User Functions, Device Functions, Scenario Functions, Report Functions, which are shown in the image 1.



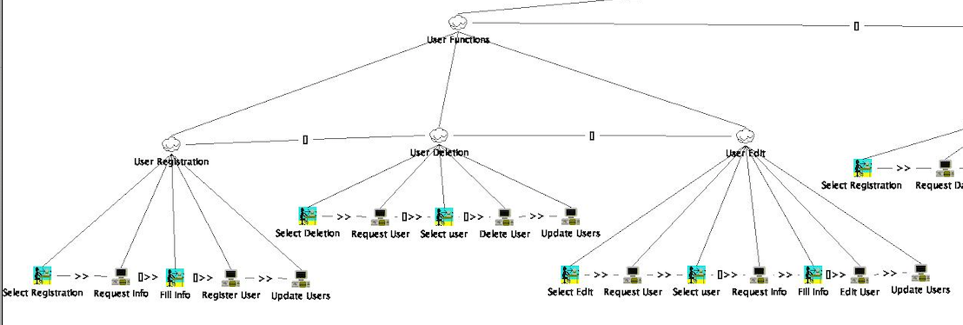
**Image 1 |** Task Model Diagram

The Enable Access workflow shows the steps the system follows to validate any user in the system, it requires the username and password in order to access to the next abstract functions. **Image 1.1**

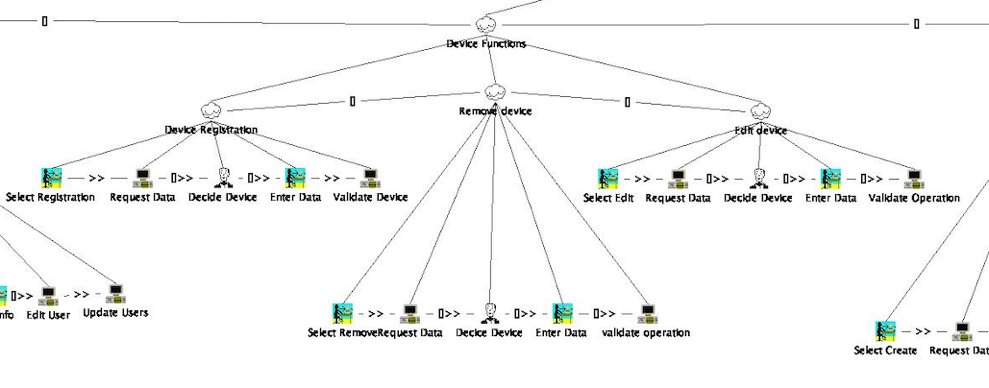


**Image 1.1 |** Enable Access Workflow

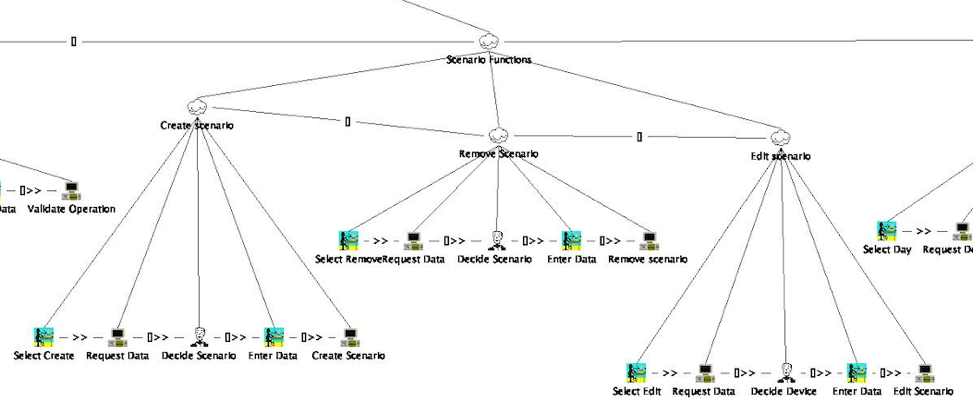
Each function in the system are classified as User Functions i**mage 1.2**, Device Functions i**mage 1.3**, Scenario Functions i**mage 1.4**, Report Functions **image 1.5** from this abstract level there’s exist CRUD functions which are explained below.



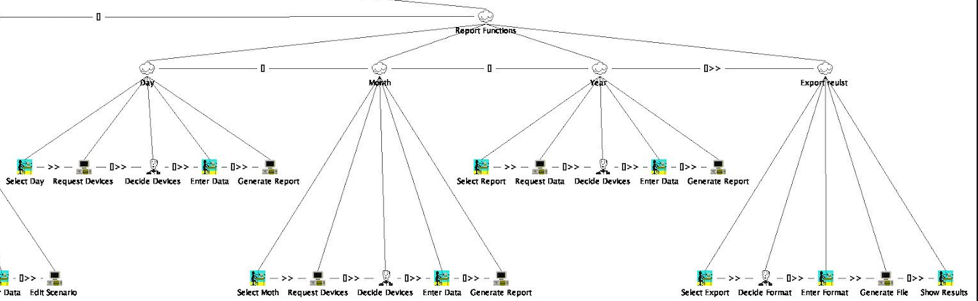
**Image 1.2 |** User Functions



**Image 1.3 |** Device Functions

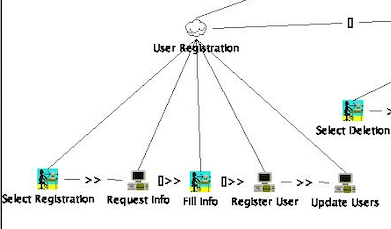


**Image 1.4 |** Scenario Functions



**Image 1.5 |** Report Functions

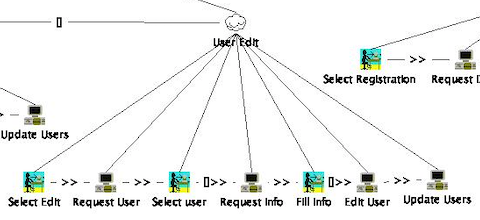
Create: It requires the information corresponding at each collection of attributes, (User, Device, Scenario, Report), then the system validates the integrity of the information in order to keep or update the information. **Image 1.6**



**Image 1.6 |** Create Workflow

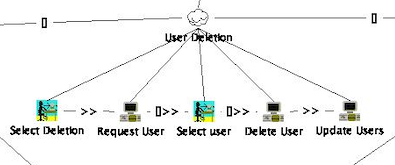
Read: The system requires a params of search, to give back the collection of attributes to show.

Edit: The system require a param of search, it give back the object information, the process to update the information is by rewriting into the data base, before of this process the system validates the integrity of the information. **Image 1.7**



**Image 1.7 |** Create Workflow

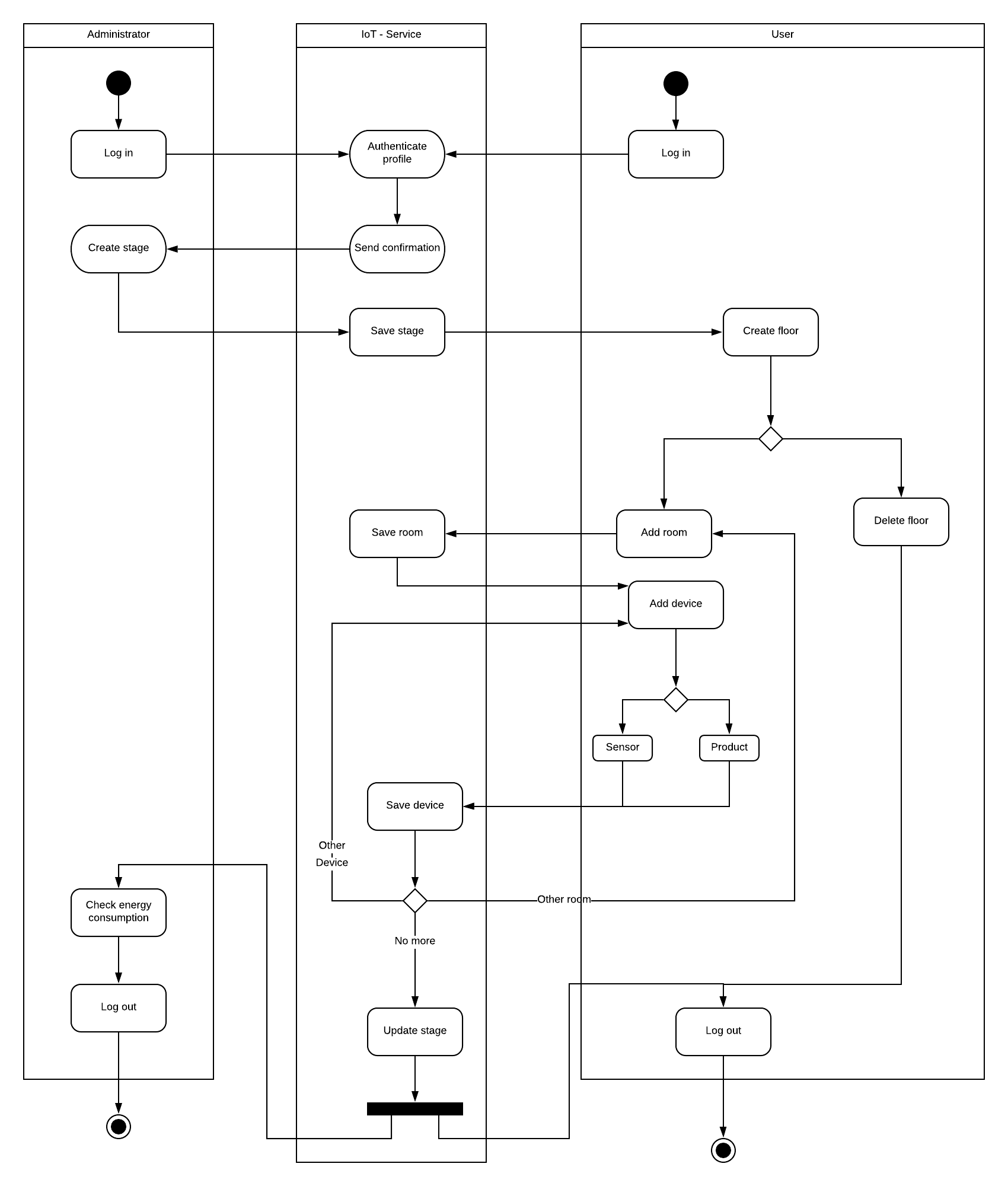
Delete: The system requires a param to search the desire object, in order to follow the Data Base Rules it is impossible to delete information form, instead we change the attributes of the object as deleted. **Image 1.8**

****

**Image 1.8 |** Create Workflow

ACTIVITY MODEL

The activity model describes the events of the system, it contains the related behaviour components and the flow between actions, in order to know which function should I do first, the **image 2** show the activity model diagram with swim lines to show the actors involved.

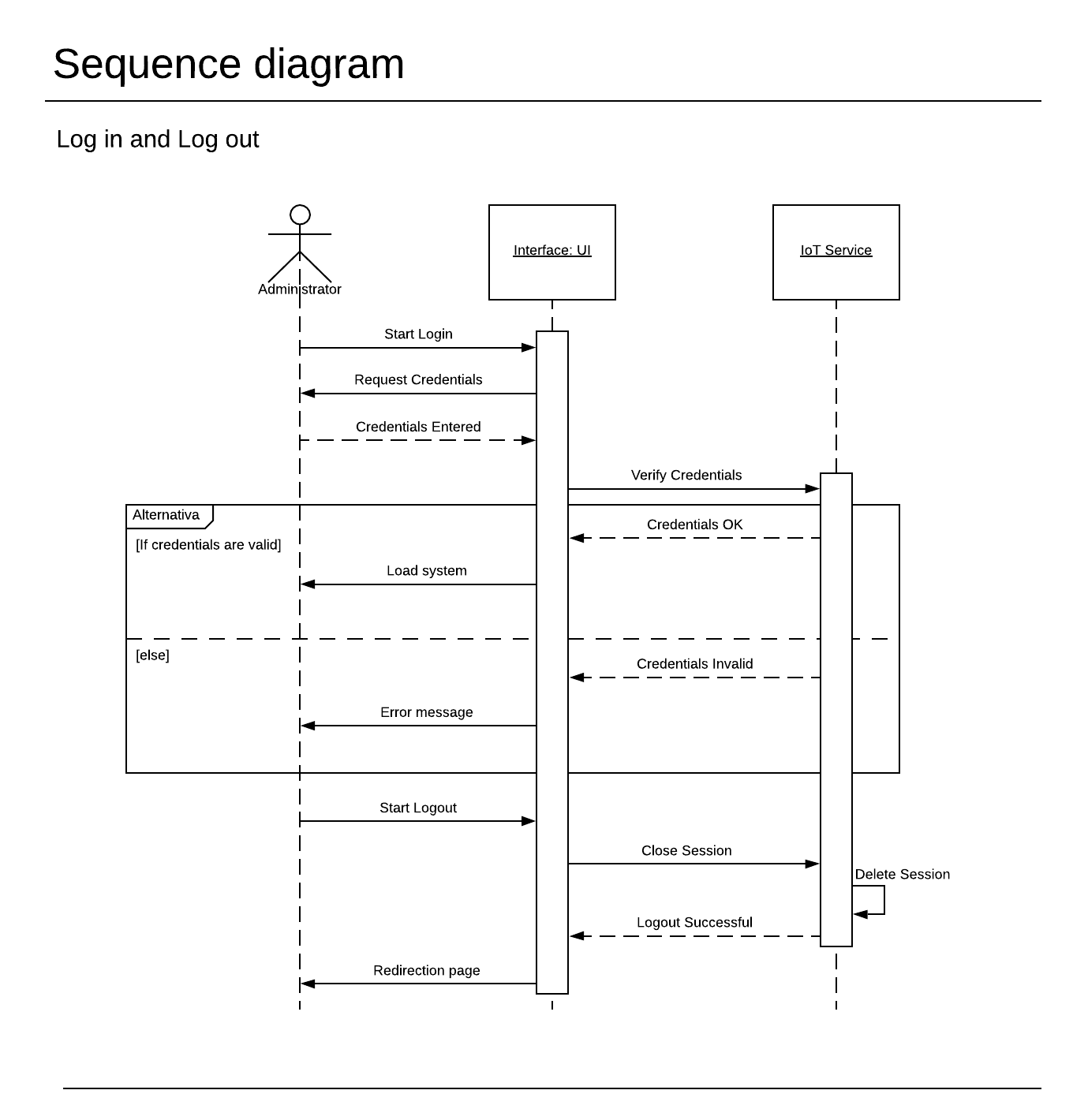


**Image 2 |** Activity Model Diagram With Swim Lines

SEQUENCE DIAGRAMS

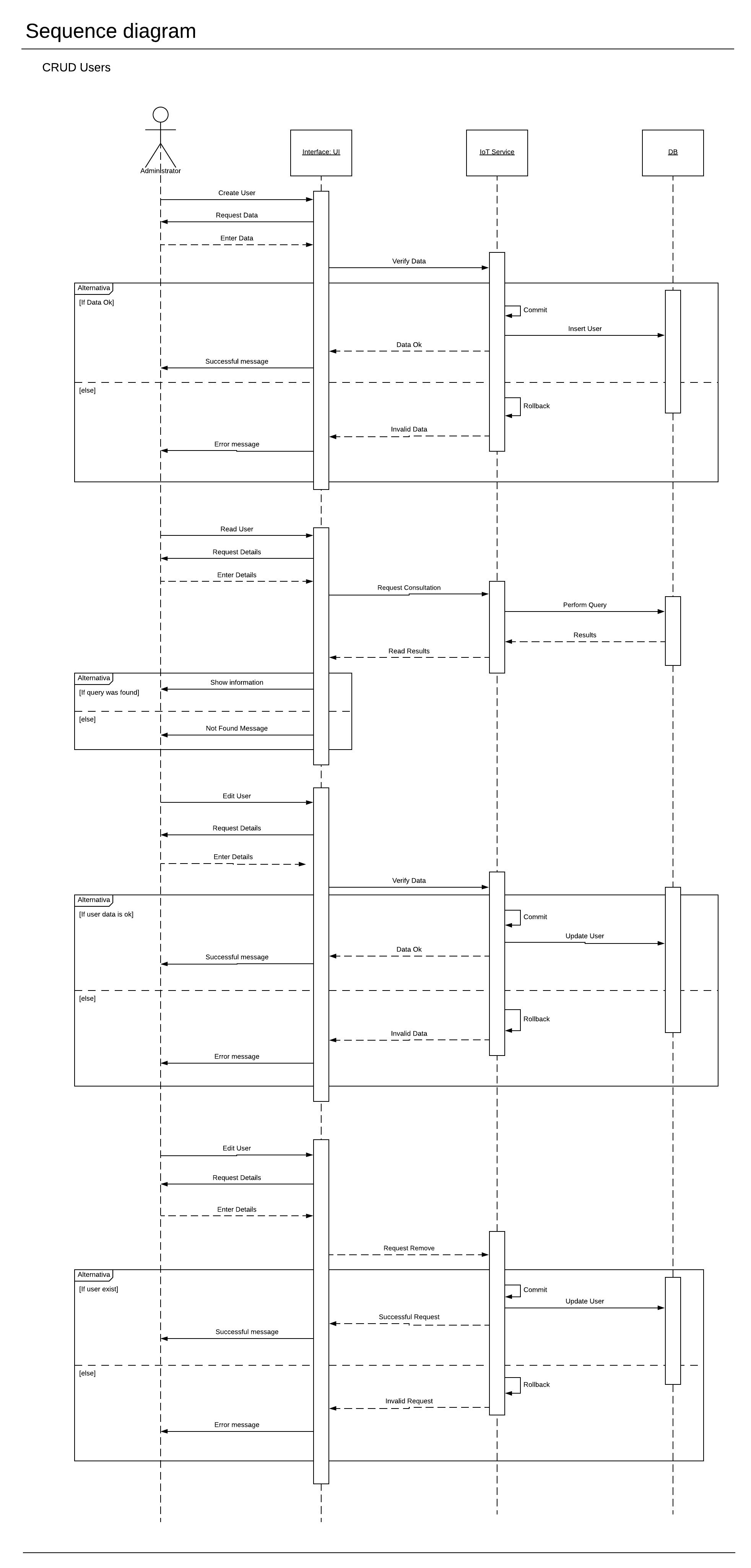
The sequence diagrams show the interaction in time sequence of each activity, the activities include are those mentioned early in the task model section.

The first sequence diagram is from the user login and logout as shown in the **image 3.1.**

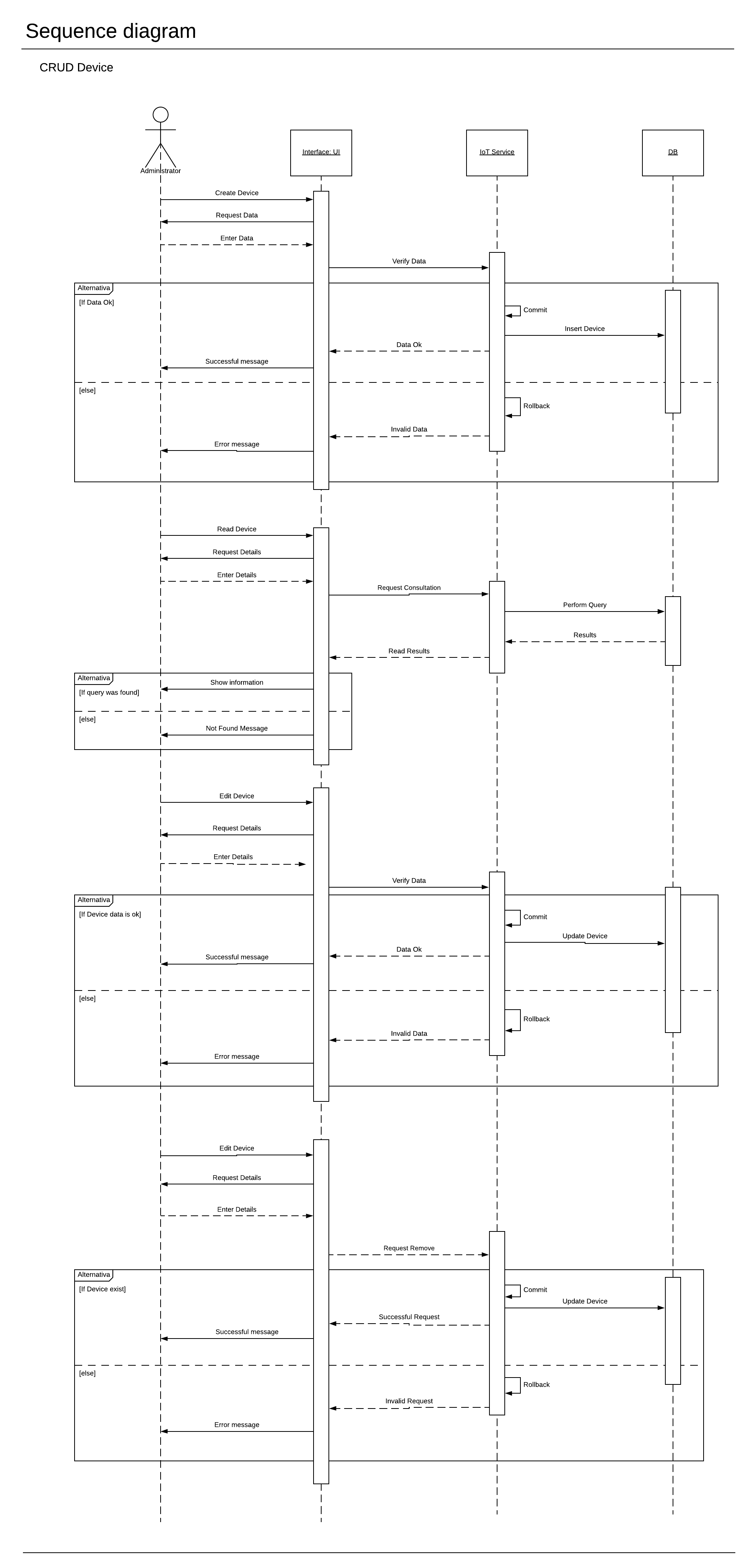
****

**Image 3.1 |** Sequence Diagram Login and Logout

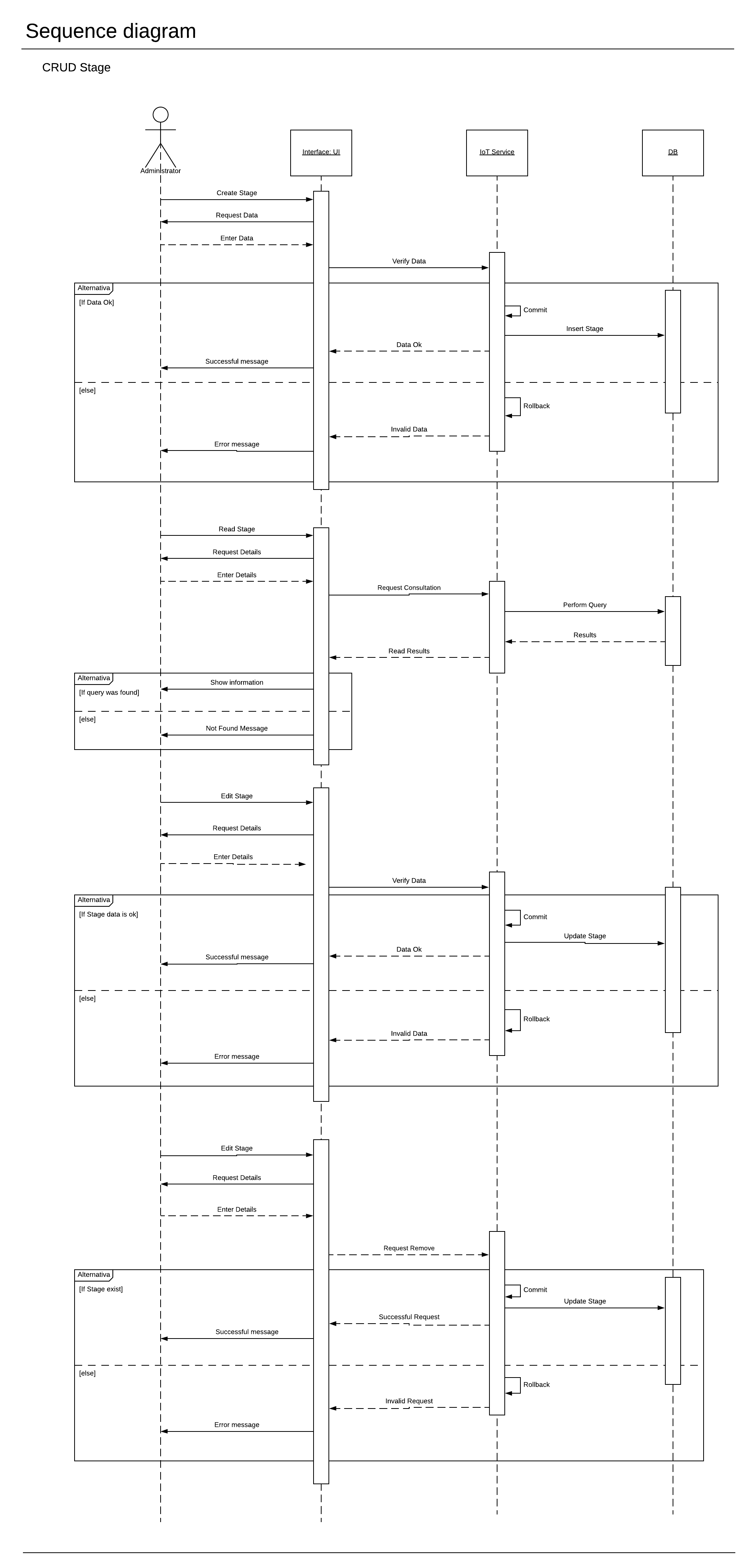
The next sequence diagrams are for the CRUD functions explained before, the User CRUD functions, the Device CRUD functions, the Stage CRUD functions, and the Report CRUD functions.



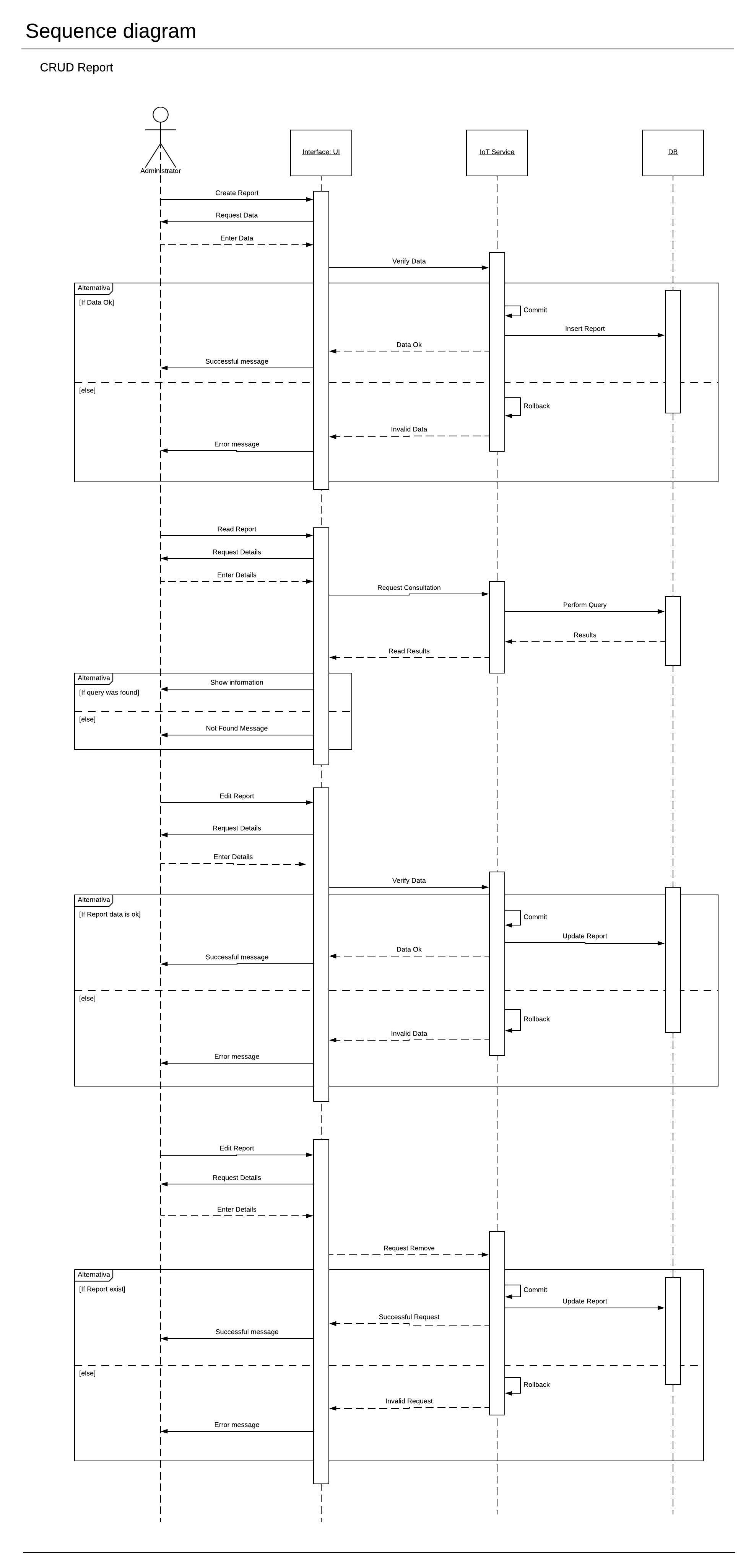
**Image 3.2 |** Sequence Diagram CRUD Users



**Image 3.3 |** Sequence Diagram CRUD Device



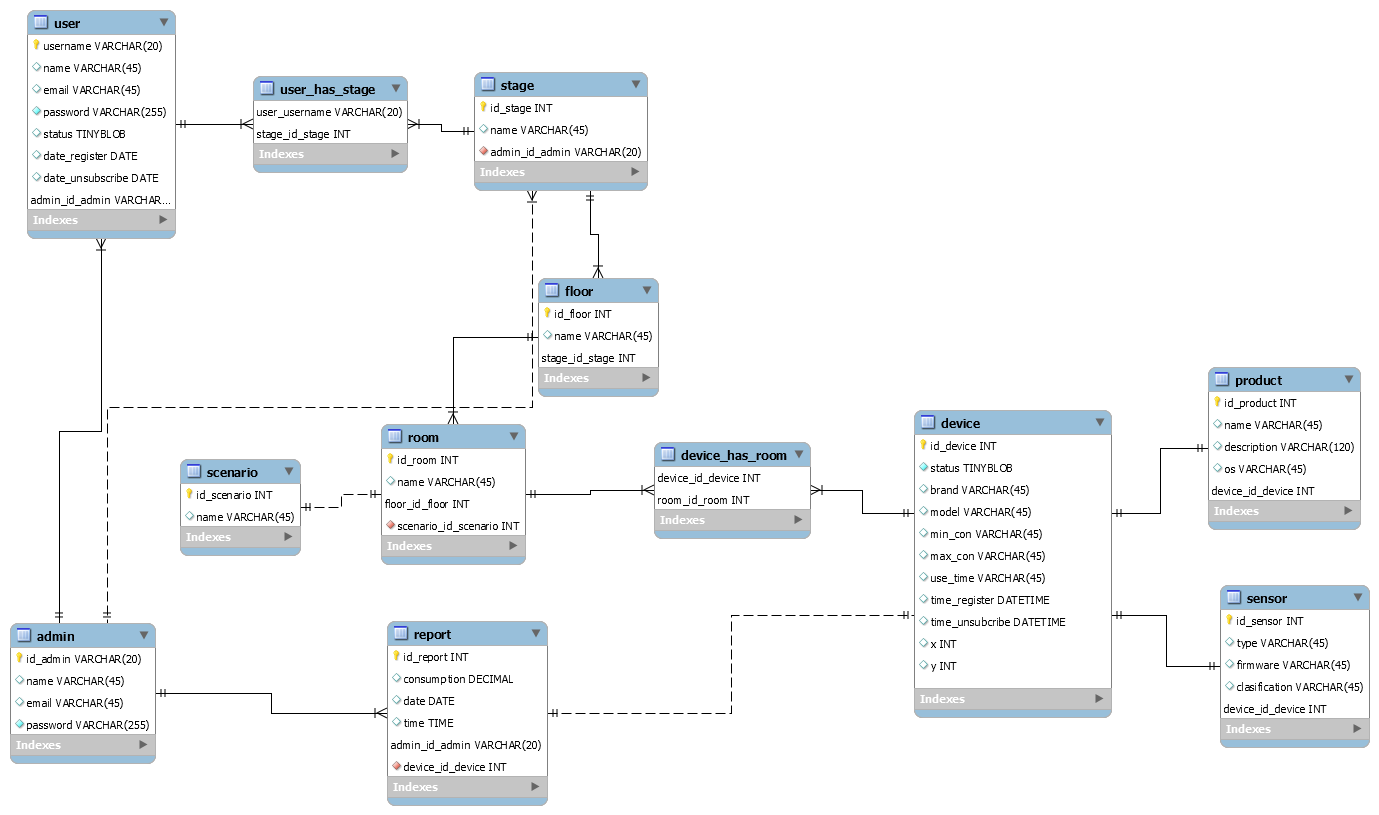
**Image 3.4 |** Sequence Diagram CRUD Stage



**Image 3.5 |** Sequence Diagram CRUD Report

DATABASE MODELING

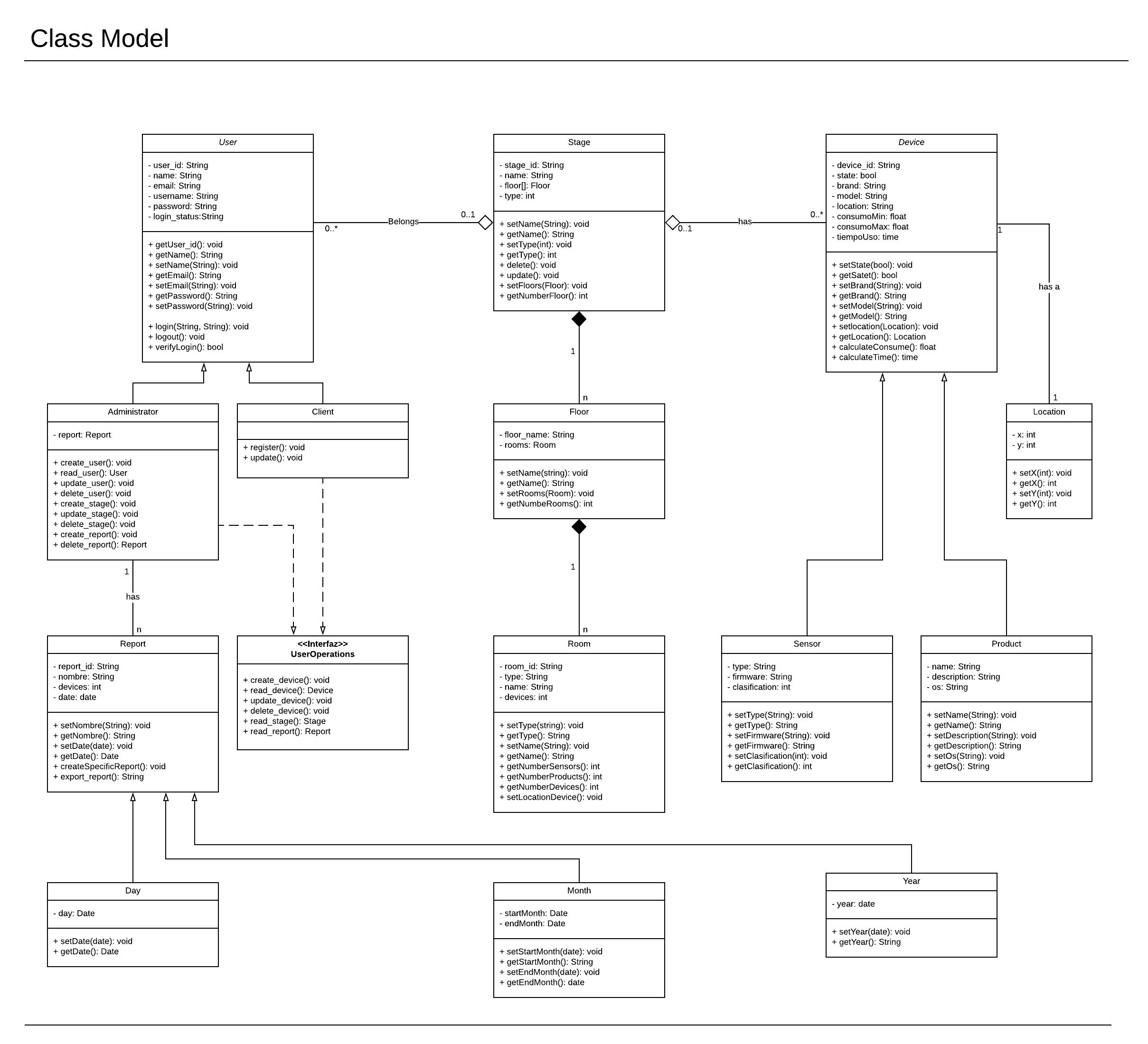
The relational model of the database, **image 4**.



**Image 4 |** Relational Data Base Model

CLASS DIAGRAM

The class diagram was made following the UML convention, **image 5.**

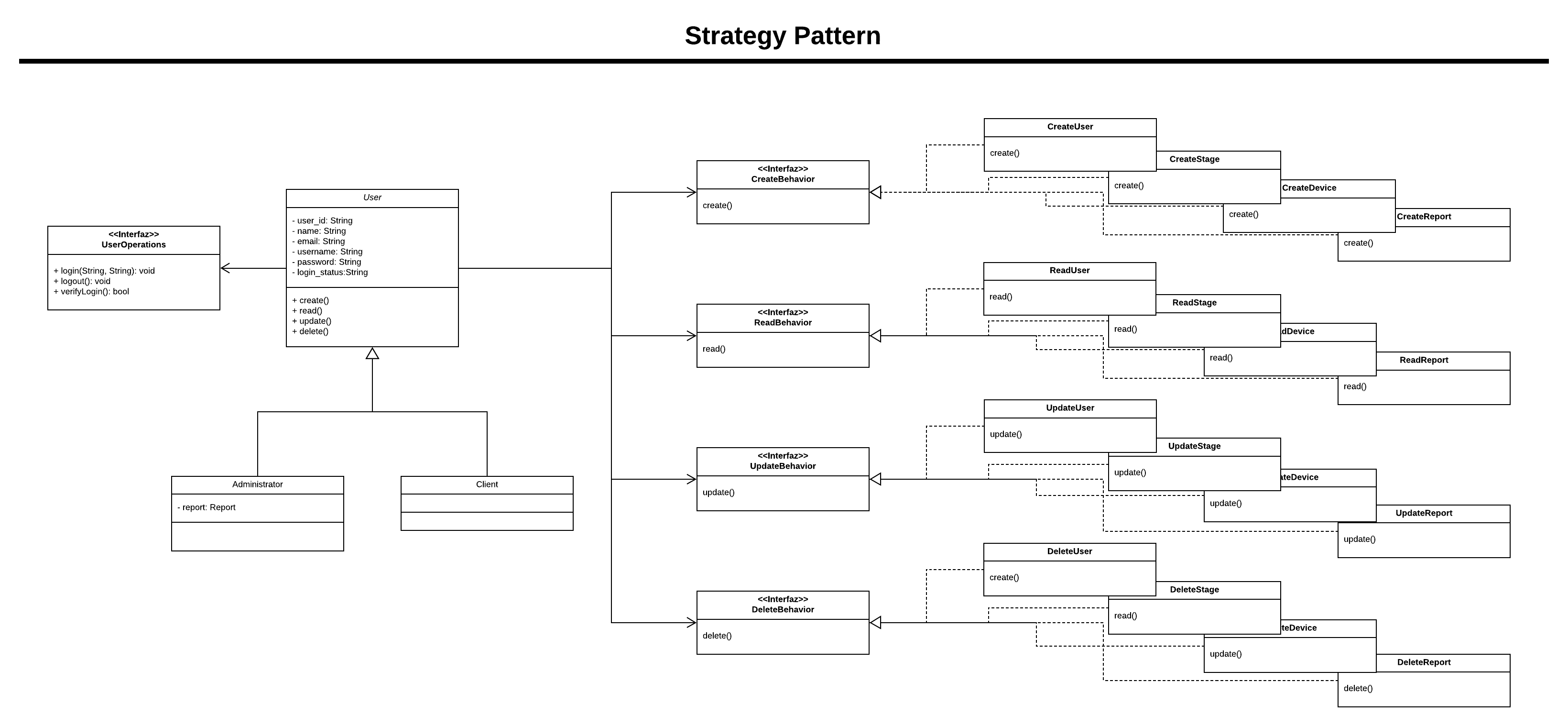


**Image 5 |** UML Class Model Diagram

PATTERNS APLIED

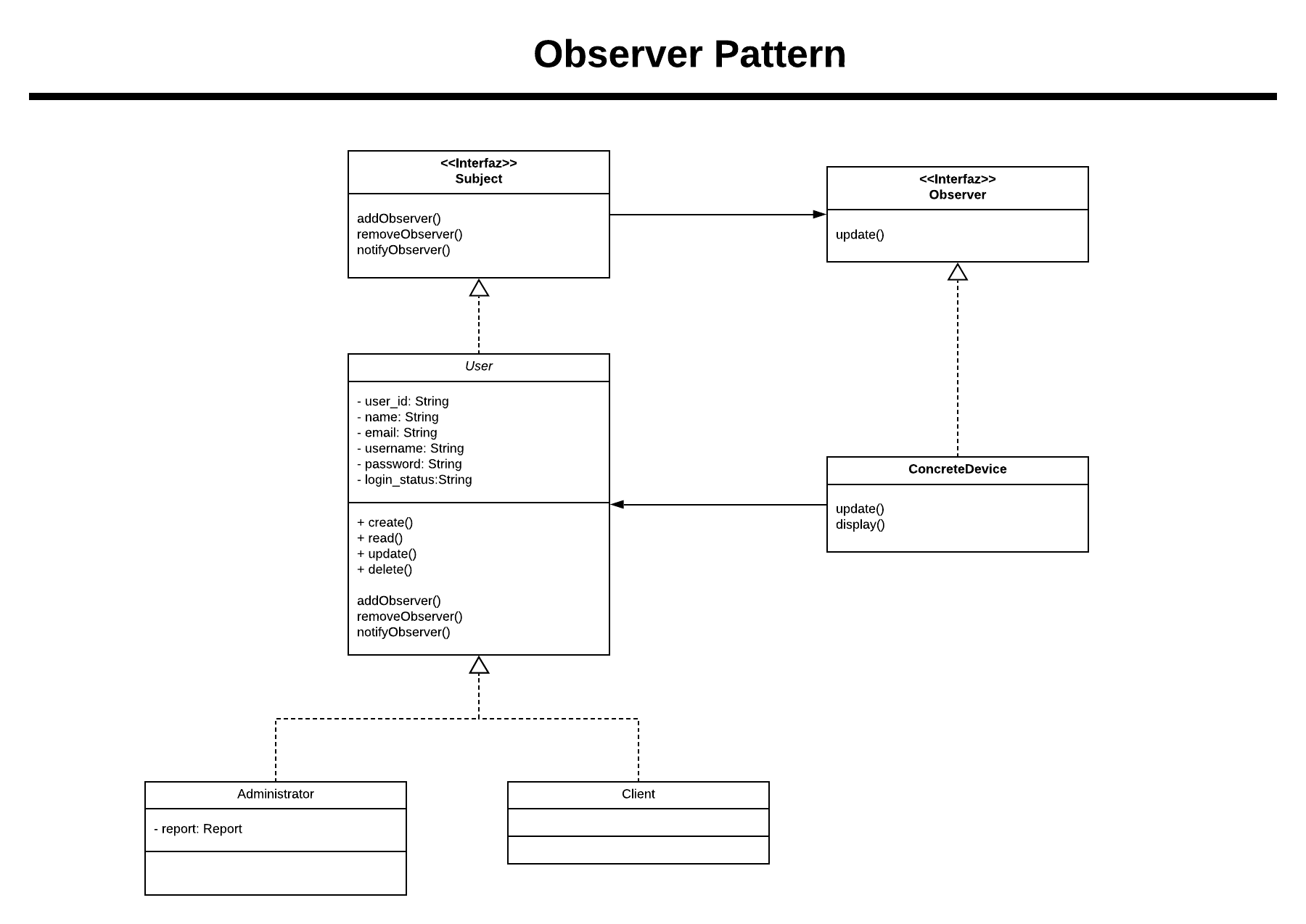
The patterns applied to the system are Strategy Pattern, Observer Pattern, Command Pattern, the corresponding diagrams are shown below.

The Strategy Pattern implements functions for every user, it has four interfaces for the control of the CRUD functions, **image 6**.



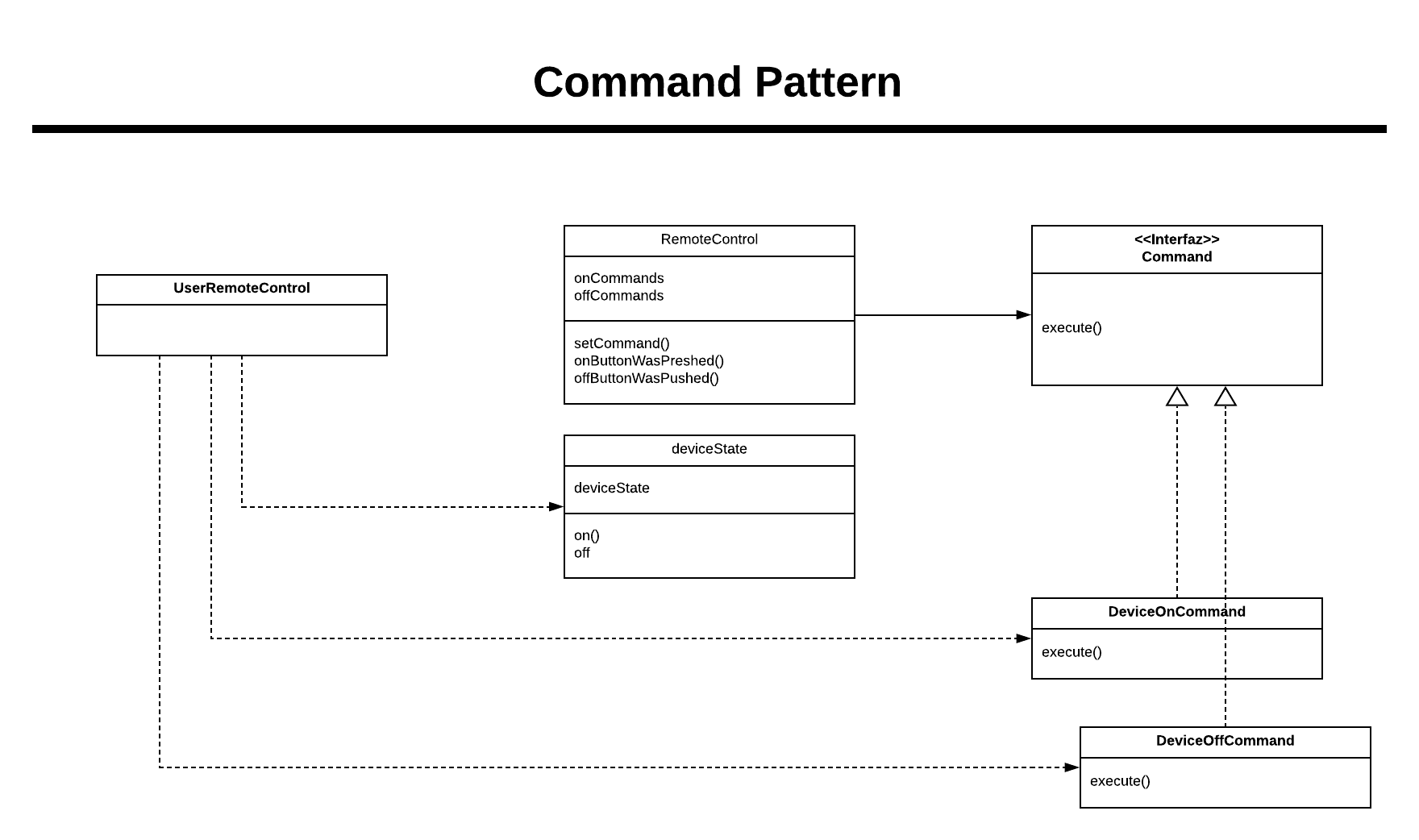
**Image 6 |** Strategy Pattern

The Observer Pattern implements functions to keep all user inform about the state of each device, **image 7**.



**Image 7 |** Observer Pattern

The Command Pattern implements functions to control the state of each intelligent, the implemented functions are turn on, and turn off, **image 8**.



**Image 8 |** Command Pattern