

REQUIREMENTS FROM PRI

The owner of a company dealing with the design of playgrounds for children plans to use an IT system to support his activity, based on the following requirements.

1. The planned system should contain information about employees of the company, about designers who may be employees but most often work on a contract and are not employed in the company and on individual clients. It is not planned to store information about other people.
2. For each employee remember: name, phone number, dates of employment; for the designer - name, surname, telephone number, list of permissions with at least one item, and for the individual customer - name, surname, telephone number, type of identity document, ID number.
3. For all customers, please remember the contact details, with the name and NIP additionally for the customer-institution. If the customer came to the company on the recommendation of another customer / client, it should be remembered which client has done the recommendation.
4. The client orders the design and construction of playgrounds (only 1 at time). A given playground is made for only one client. The data of potential customers is not stored. If the customer orders a playground, the contract data should be stored, including: the date of conclusion, planned date of commencement of implementation, planned date of completion, the actual date of completion of the playground, amount of down payment (optional), contract status ("concluded", "implemented", "canceled", "completed").
5. For the playground store: address, description of land, surface, is it fenced, fence height. The entire playground, during design, is divided into zones for which the surface was determined, the degree of insolation (on a scale of 0-3, where 0 means no insolation), and the type of substrate.
6. The system stores information about devices in the company's offer. Each device is described by catalog number, name, security certificate, a list of the material, with which it is built, and the list of images, min., and max. age of the child, which is intended for the device, and the type of substrate to which it can be installed.
7. A device can be installed in a selected zone/zones of the playground, as long as the ground type agrees. You can install any number of devices in one zone.

8. All devices are divided according to the type, among others, into slides (the length of the exit is given in cm, and the angle of fall is known). The division is complete.

9. Each playground is designed by one designer, while currently implemented by at least 2 employees, but no more than six (it is possible to change these values in the future). During the implementation of the playground, Employees are assigned identifiers with a specific expiry date and a unique number on the given playground.

10. The system should support the company in the implementation of functions such as:

- Displaying the list of offered devices (guest, customer, owner);
- Display information about the selected playground (customer, owner);
- Displaying the list of employees assigned to a given playground (owner) - also possible while displaying information about a given playground;
- Creation of a set of playgrounds completed during a given period (owner) - along with a list of employees assigned to a given playground ;
- Introduction of a new device, where the algorithm depends on the type of device (owner);
- Removal of canceled contracts (automatically, half a year after cancellation).

USE CASE DESCRIPTION

From functionality suggested in the PRI part, it will be the introduction of a new device that will be the subject of this document. Other functions are also described in the class diagram but won't be given as detailed explanation. The programming language chosen to implement the solution was C#.

Display is handled by the program client with an interactive GUI, made using WinForms. Data is fetched from the database, via linq instructions, and displayed using data grids. There are two "tabs" for devices and playgrounds used to get corresponding tables. In the device tab the devices of requested type are displayed after making a choice using combo box. After clicking in cell representing a particular device a new form with a list of used materials is shown. Additionally images of desired device can be shown after pressing the correct button.

Since devices require a list of materials they were constructed from to be stored, there was a need to add additional Materials table joined via junction table with the Device table representing a many to many relationship. The list of materials is fetched using associations in that junction table.

Devices can be of many types, an inheritance mechanism was necessary. The type of structure here chosen here was Table Per Hierarchy. Since all devices share most fields with few discrepancies, TPH seemed like a good choice. This method is also best for filtering through records as it requires no joins and this type of data will be filtered often.

The images in databases aren't always stored directly, as byte for example, and can be instead represented as links to a server hosting them. Since the application has no connectivity to any server the images are stored as a path to project files where the images are stored.

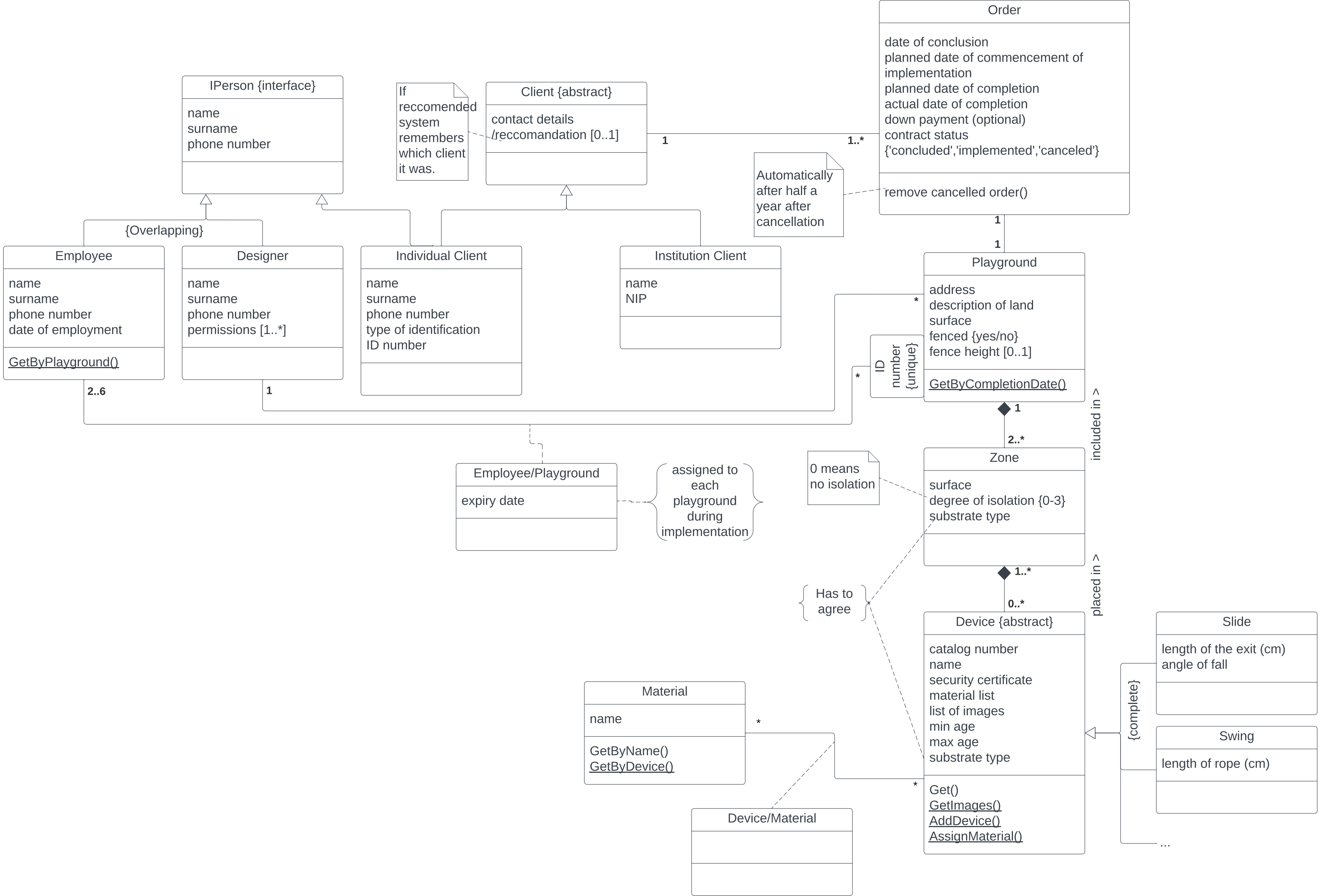
When add button is pressed a new form appears. In this form user is able to create a new device after correctly filling in all fields. Upon confirming the choice the program creates a device object and a list of materials used to make the device. This new device instance is then passed further to query the database to add it into the records. After successfully adding the device an association between materials and said device is made by further querying the database. Any problem during this process results in a transaction being rolled back, and is only committed after everything is set up correctly.

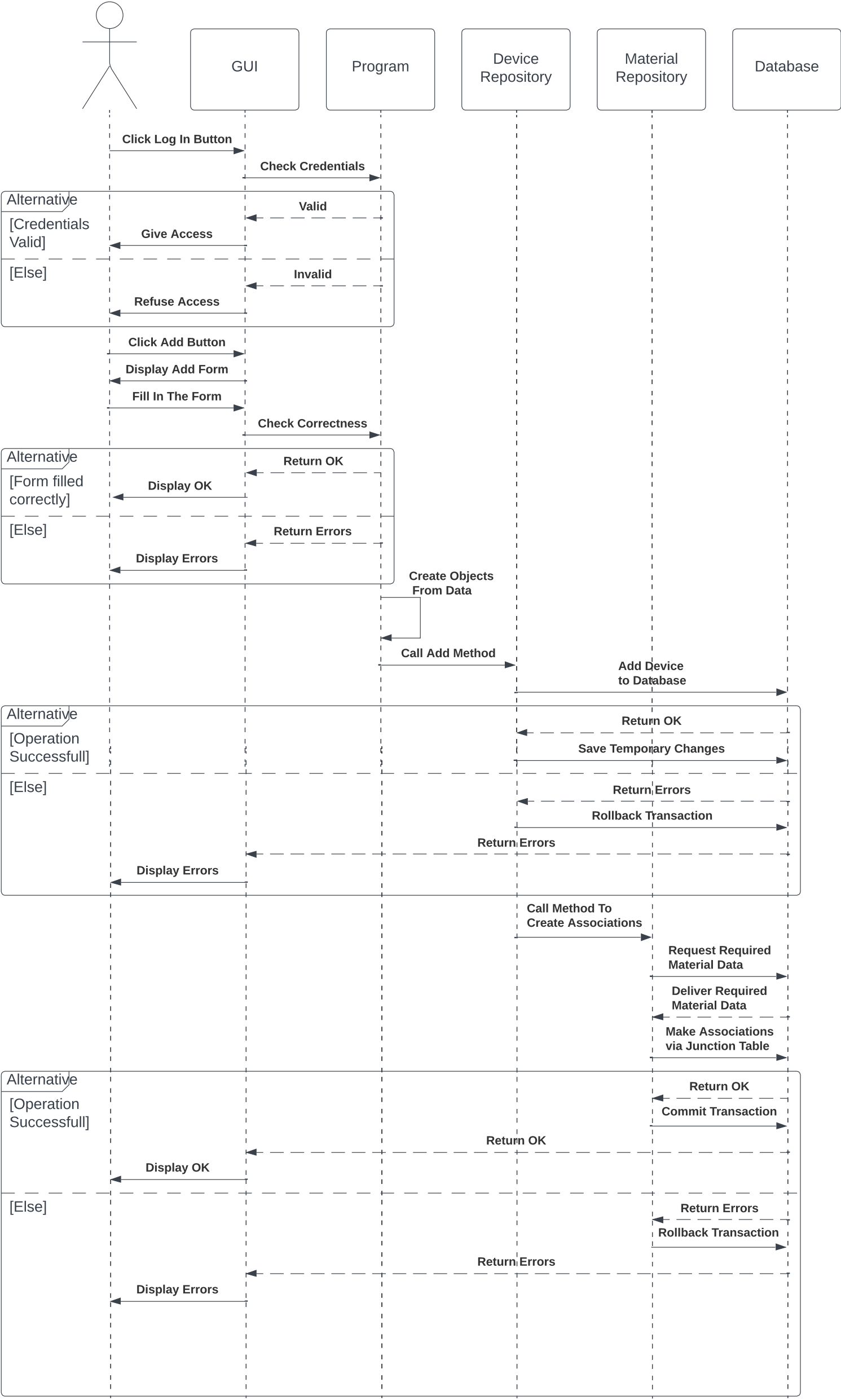
The playgrounds can be filtered based on date of completion using a graphical calendar when pressing a right field. The filter fetches data ever time it's changed sacrificing some performance for more dynamic and up to date data representation. From playground tab associated employees can also be viewed. Similar to materials, here there is also a junction table with an addition to expiry date and an ID generation that is unique to an instance to assigning an employee to a playground.

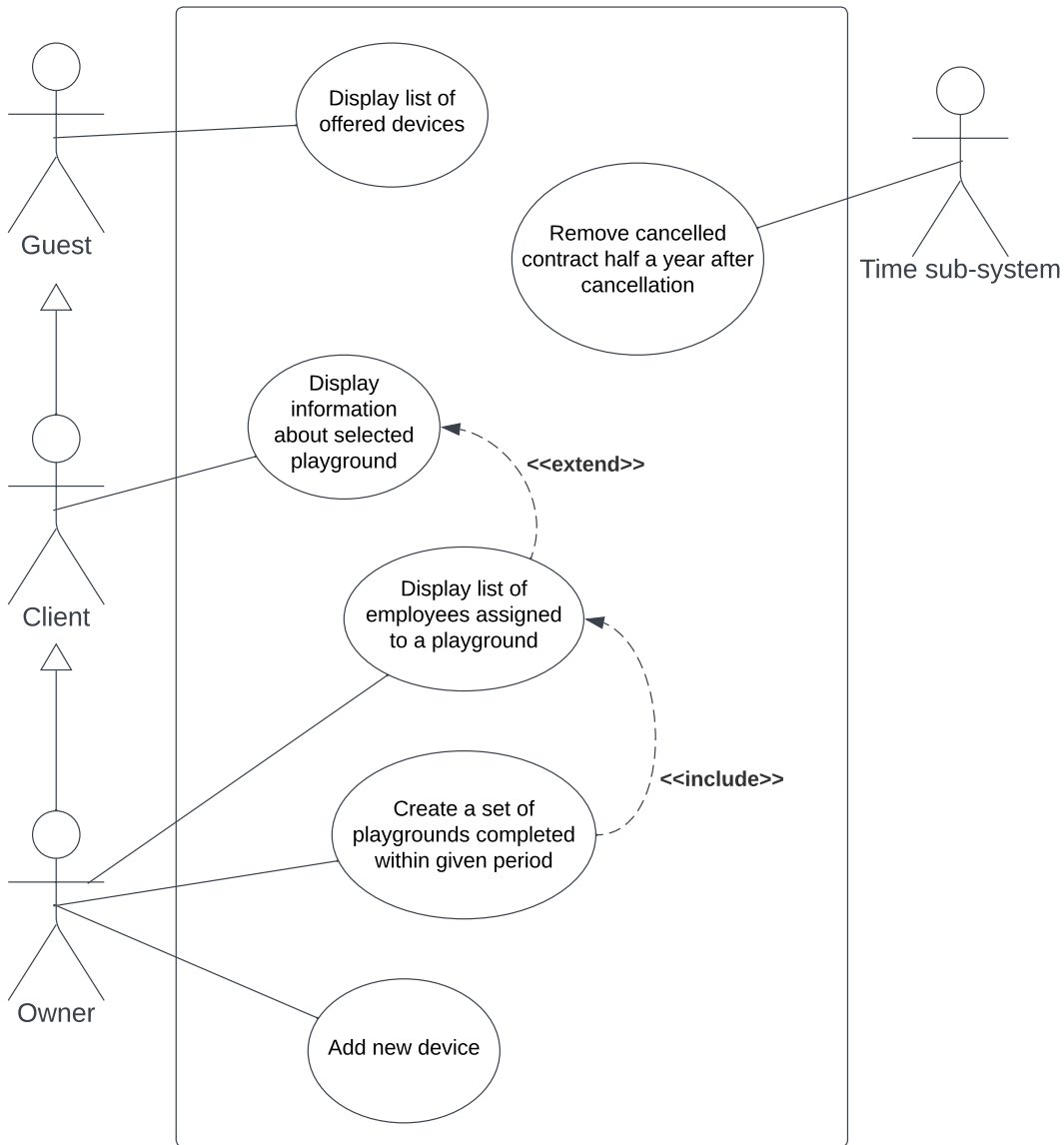
Another inheritance challenge was represented by an Employee and Designer association – overlapping inheritance. This issue was resolved by giving the Designer model a nullable Employee field, giving a Designer that is also an Employee access to Employee methods through that field.

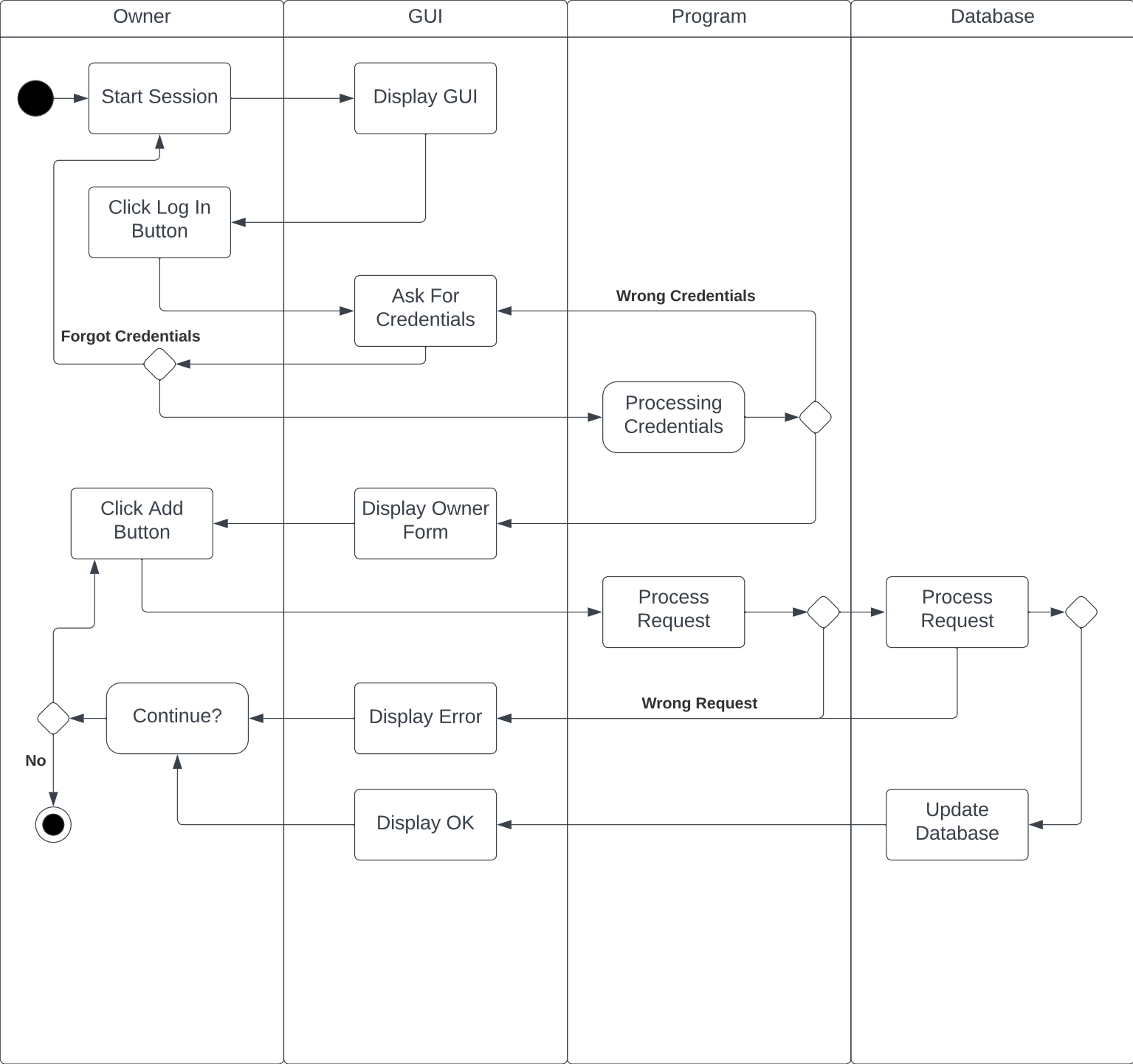
Since Employee, Designer and Employee all share similar fields they all implement IPerson interface.

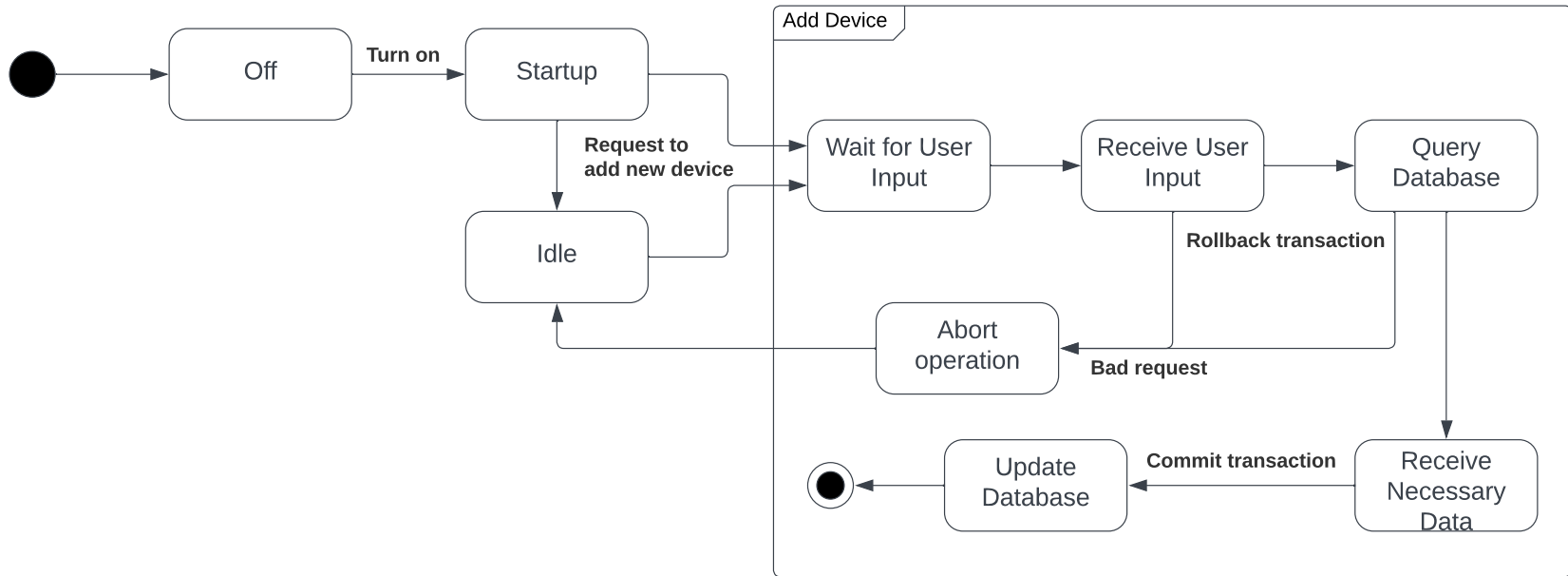
As mentioned earlier there are two types of clients – individual and institution. This was implemented using Table Per Type structure. Both the inheriting classes are distinct enough that there is basis in using this solution. Most queries are performed based on client ID, so having one parent class here is sufficient. If there is a need to perform operations on particular type of clients having distinct tables can be helpful here. Compared to TPH, TPT has more querying problems between classes due to the need to use joins, but offers a cleaner less redundant representation.












Starting screen:

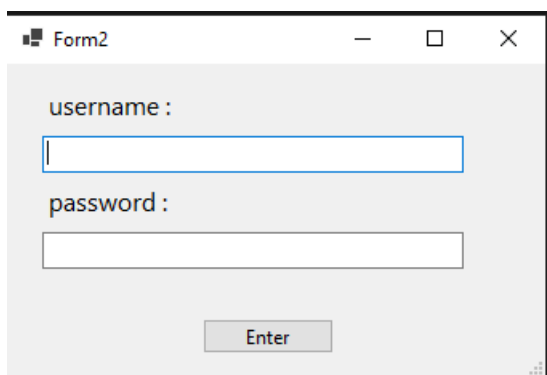


The screenshot shows a window titled "Form1" with a standard Windows title bar (minimize, maximize, close buttons). The main content area has a light gray background. At the top left, the word "Offer" is displayed in a large, dark font. At the top right, there is a "Log In" button. Below the "Log In" button is a dropdown menu labeled "Pick Type" with a downward arrow. The central part of the window is occupied by a large, empty gray rectangular area. At the bottom right, there is a button labeled "images".

The main form is resizable with elements being anchored as needed, the other form are of fixed size and must be closed before interacting with the main form again.

The elements visible here are based on who is logged into the system. Here is the view of the owner – as the person with access to adding new devices.

But first a log in is required.



The screenshot shows a window titled "Form2" with a standard Windows title bar. The main content area has a light gray background. It contains two labels: "username :" and "password :". Below the "username :" label is a text input field. Below the "password :" label is a text input field. At the bottom center, there is a button labeled "Enter".

Form1

owner

Log Out

Devices

Playgrounds

Slides

| | CatalogNum | Name | SecurityCerti | MinAge | MaxAge | Materials | lengthOfExit | angleOfFall |
|---|------------|-------------|---------------|--------|--------|-----------|--------------|-------------|
| ▶ | 1 | Basic Slide | cert | 8 | 14 | | 50.00 | 50.00 |
| | 2 | Twist Slide | cert | 10 | 14 | | 200.00 | 35.00 |
| | 7 | teset | sef | 3 | 4 | | 12.00 | 40.00 |

Add

images

The addition form displays errors individually for every field to make troubleshooting easier.

AddDeviceForm

Add device

Name:

error

Type:

Slide

Length of exit

qwer

Invalid value

Minimum Age:

3

Security Certificate:

Invalid certificate

Angle of fall

3

Maximum Age:

max

Invalid age

Substrate:

Sand

Materials

☐ Plastic

☐ Wood

Pick material

Add