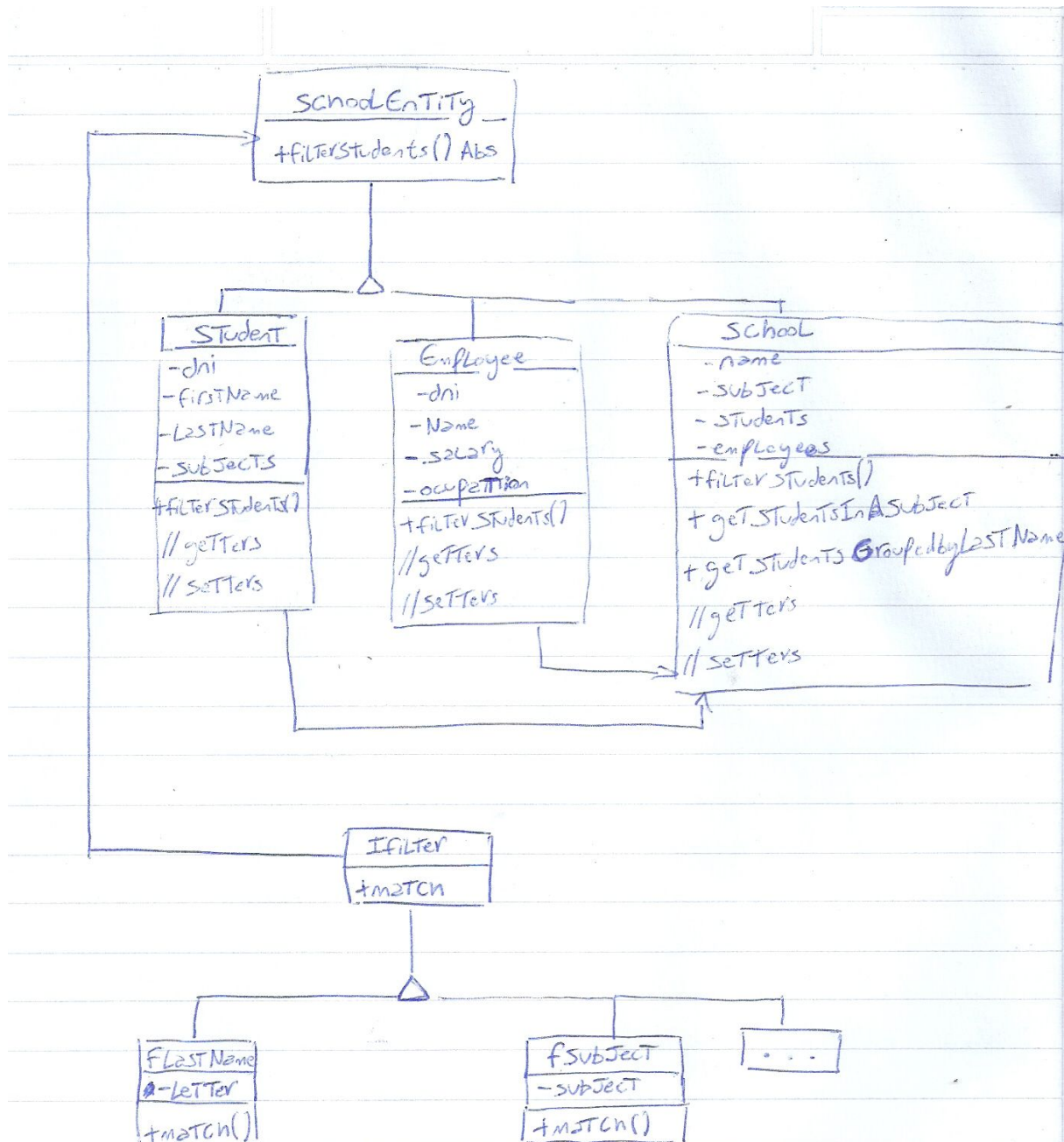


## Exercise One

Create a domain model implementing the concepts of hierarchy, having as example a school's structure where we find different types of people.

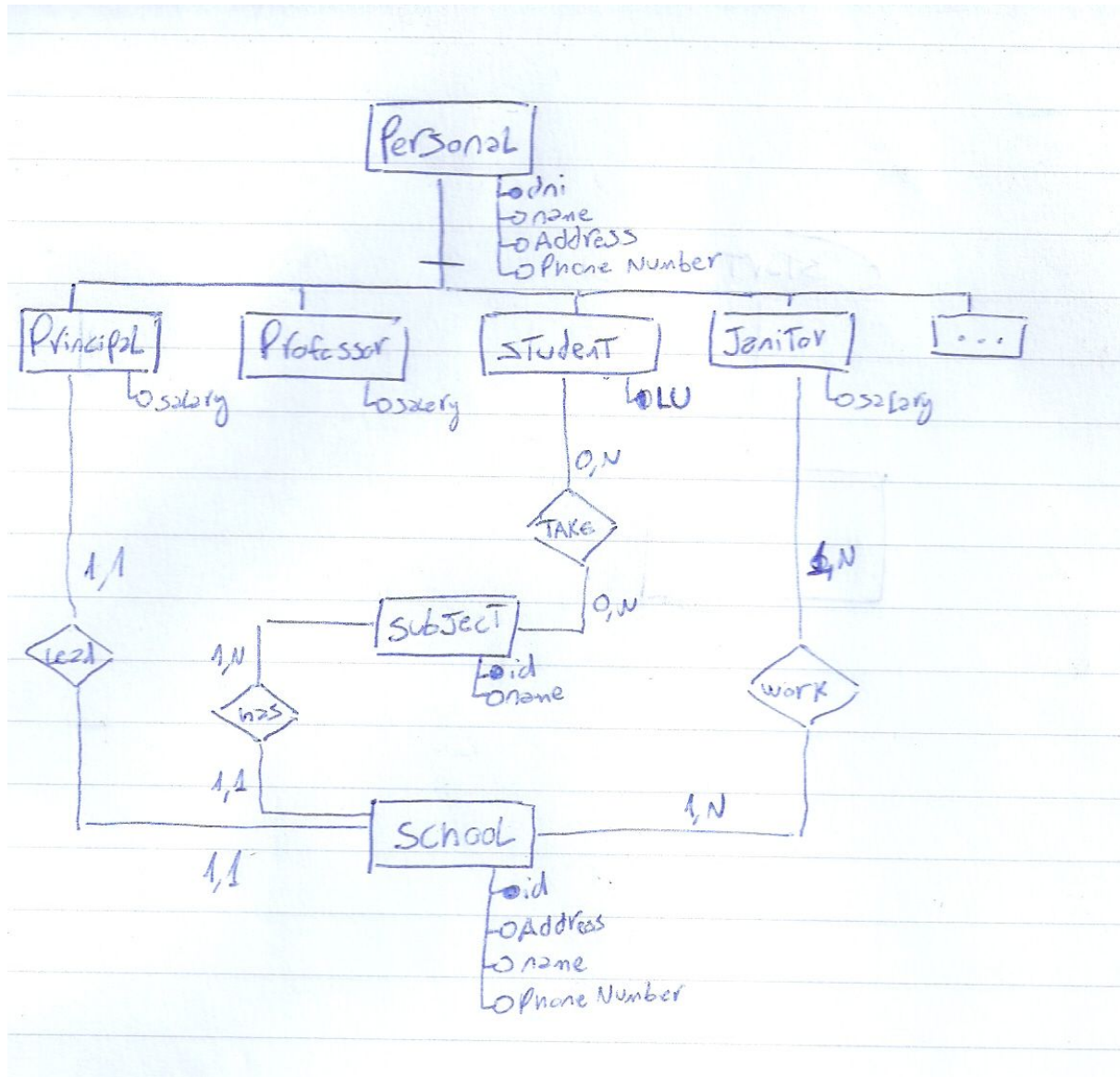
Examples of possible entities are: Principal, Employee, Student and Janitor. Try to implement the concepts of Abstract class and Interface.

Algunos atributos fueron delimitados simplemente como del tipo String ya que pueden ser simplemente diagramados como clases POJO, como es el caso de Subject, o de Room. No se realizó de este modo ya que aun así, esto no afectaba al ejercicio en sí.



**C) Create different database structure diagrams with the possible implementations of the created domain.**

Varias entidades están abiertas a la adición de más atributos, como puede ser e-mail, en la entidad personal. Como también se puede agregar nuevas entidades dependientes de la jerarquía "Personal", según se lo necesite, como podría ser la entidad *Preceptor* o *Tutor*.



**D) Consider that I want to query all the people in the database and the engine is taking too long to respond. Which are the possible solutions? How would you determine the problem?**

Primero se deben traer solos los datos de las personas que se necesiten. Realizar un **Select \*** sobre una tabla es muy costoso, y no es conveniente realizarlo (a menos que realmente se lo necesite). Por otro lado, realizar una paginación de la tabla acelerará el proceso de consulta, ya que se comenzará a traer conjuntos separados de filas, evitando así que se tenga que traer toda la tabla junta de una sola vez.

**E) I want to query all the students which ages are between 19 and 22. How would you optimize this query?**

Para realizar esta consulta, indexaría esta tabla por edad, utilizando para crearla, la fecha de nacimiento de las personas. También sería más útil agrupar por un rango de edad, lo que aceleraría aún más la consulta. Utilizando índices, nos evitamos realizar una búsqueda exhaustiva sobre todos los valores de la tabla. acortando así los tiempos de la consulta, sobre todo cuando el volumen de datos es grande.

**F) I want to delegate the saving process to the database engine instead of having it on the application side, which database utility would you use?**

Para poder separar el proceso de salvado de la base de datos fuera del lado de la aplicación se pueden utilizar stored procedures. Esto permite que las operaciones no sean visibles desde la aplicación, sino que están encapsuladas y ocultas en procedimientos ya compilados en la base de datos, solo teniendo que realizar la invocación de estos procesos desde el lado de la aplicación.

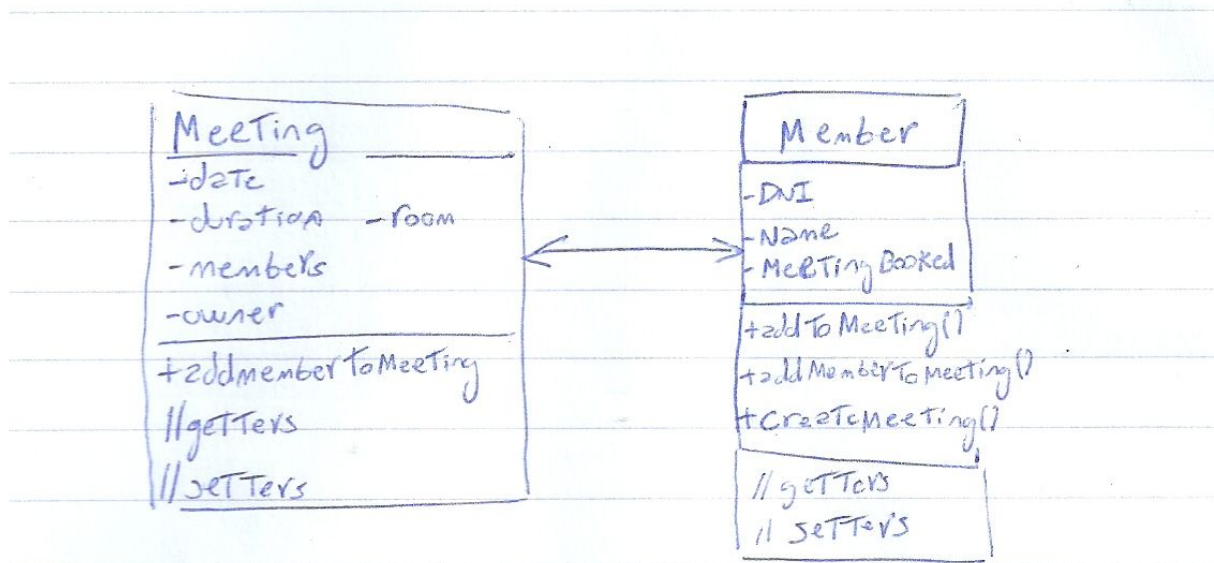
## Exercise Two

Create the domain model of a conference room schedule where meetings are booked. Bookings should contain a list of participants and only the organizer can modify them.

Incorporate all attributes and methods you consider needed for each domain entity. Have into account that bookings cannot overlap and cannot last less than 15 minutes and no longer than 3 hours.

Create a diagram of how the user interface would look like and how a booking flow should be.

Class Diagram



La siguiente imagen muestra una maqueta de como podría ser la interfaz de usuario de el scheduler requerido.

Meeting Booked Teracode (John Johnson)

Scheduler

◀

May 10th, 2019

▶

📅

+

Add Meeting

📅+

Monday

Tuesday

Wenesday

Thursday

Friday

Saturday

9:00

9:30

10:00

10:30

11:00

11:30

12:00

12:30

13:00

13:30

14:00

14:30

15:00

15:30

Meeting 3

Meeting 2

Meeting 1

Partners

👤

Juan

👤

Carla

👤

Pedro

👤+

Add member to the meeting

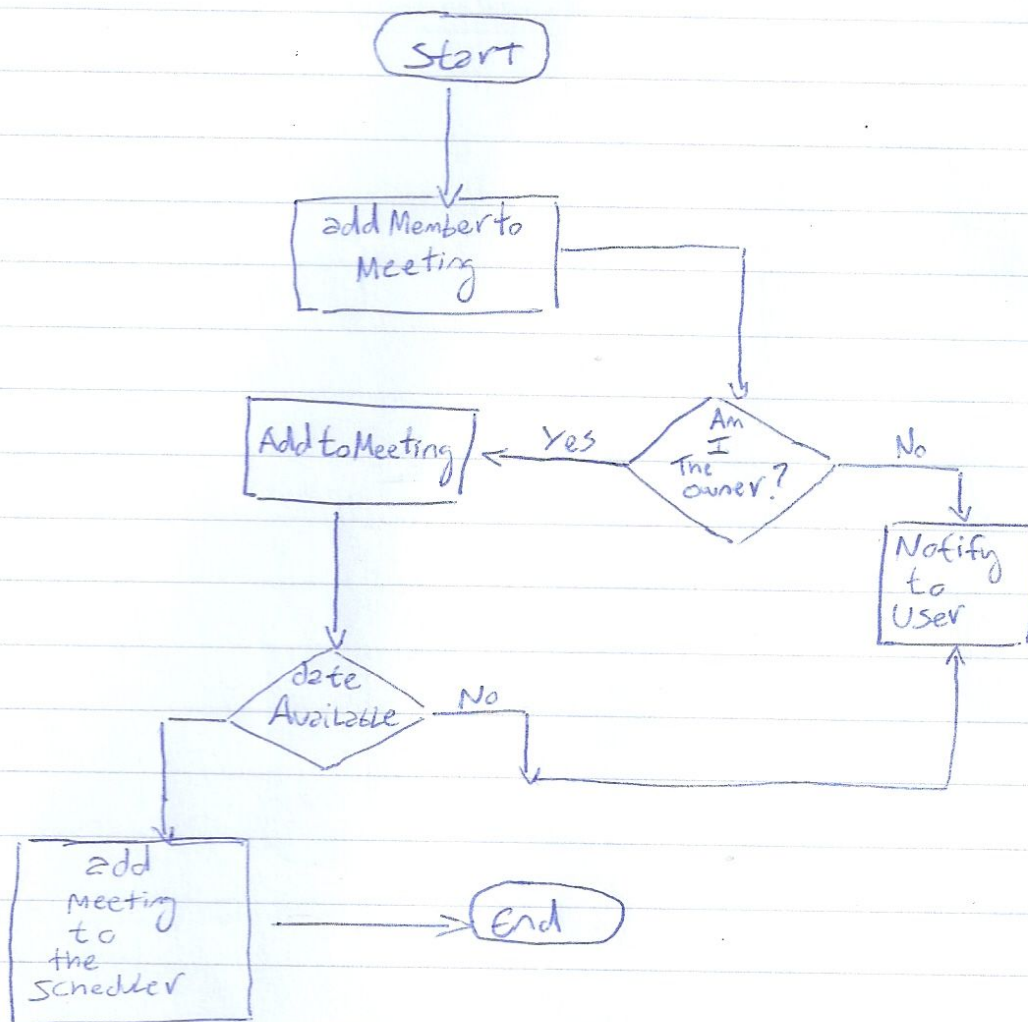
Participants

John (me)

Jane

Los siguientes diagramas de flujos son para la creación de reuniones y la adición de miembros a las mismas. La necesidad de una *eliminación* de miembros de las reuniones y las reuniones en sí es fundamental, pero al ser inversa y análoga a la creación y adición de miembros, no se creyó necesaria de implementar.

## Add Member to Meeting





## CREATE MEETING

