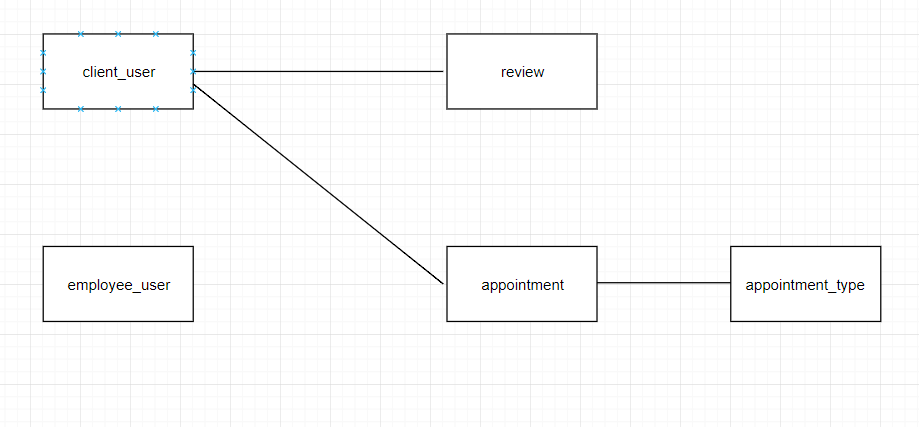
Saloon for Dogs

Analysis and Design Document

# Elaboration – Iteration 1.1

# Domain Model



The employee doesn’t communicate to anyone because he can anyway handle all the data.

# Architectural Design

## Conceptual Architecture

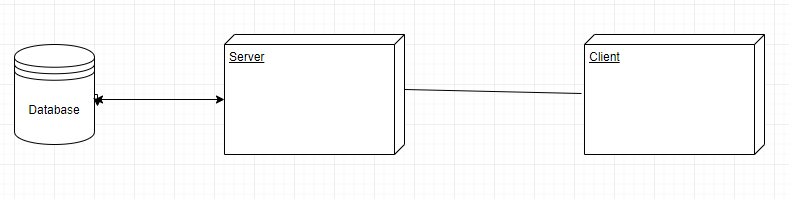
I will use

* layered architecture because I want each layer to have a single responsibility.
* Domain logic pattern: service layer: to encapsulate the business logic.
* Repository pattern because I worked with it in the 1st assignment and I understood it. It is useful because the business layer will access directly the entities it needs, without looking at how they were made.
* Transactions scripts: service per operation (encapsulates the logic related to a given database table)

## Package Design

## 

## Component and Deployment Diagrams

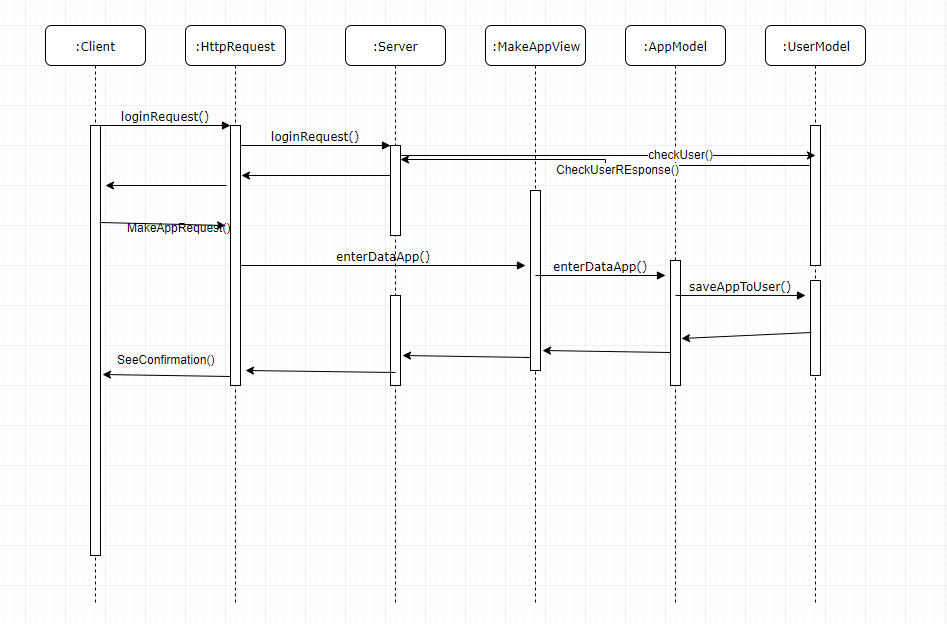


# Elaboration – Iteration 1.2

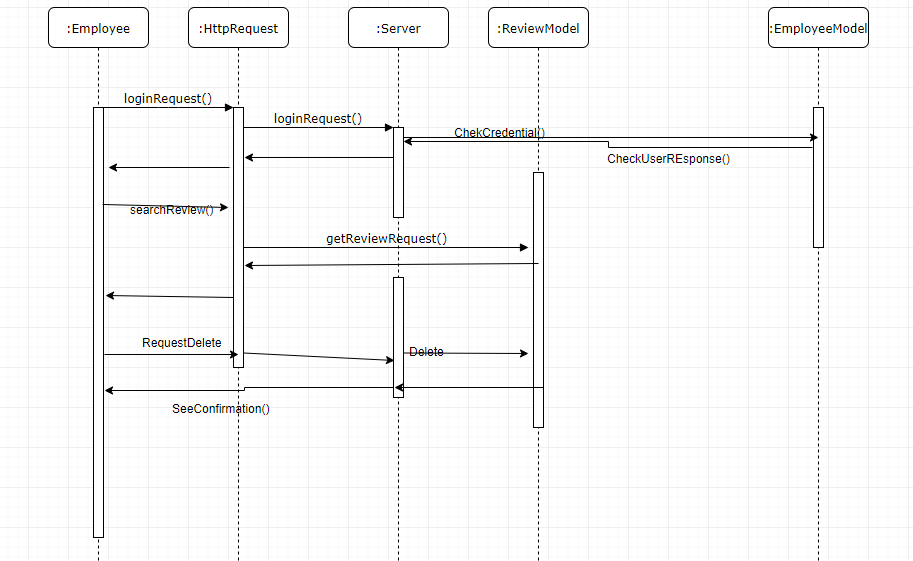
# Design Model

## Dynamic Behavior

Client authenticates and makes an appointment



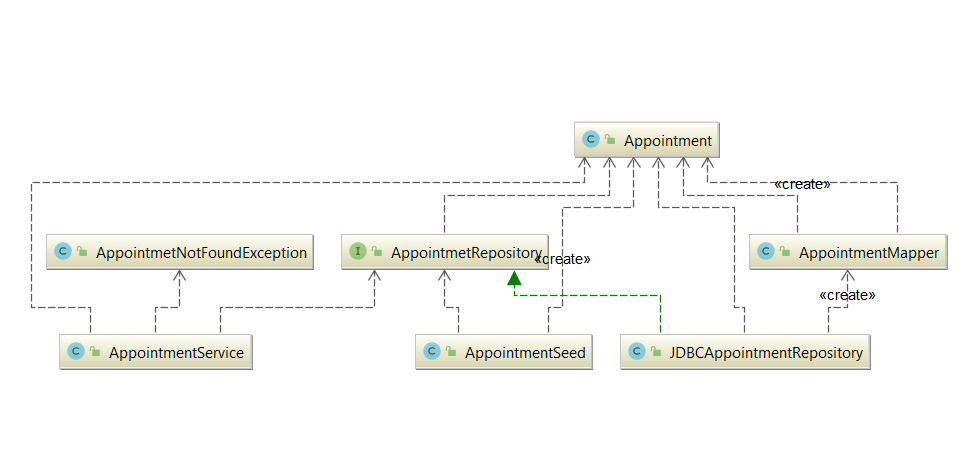
Employee authentication and delete review

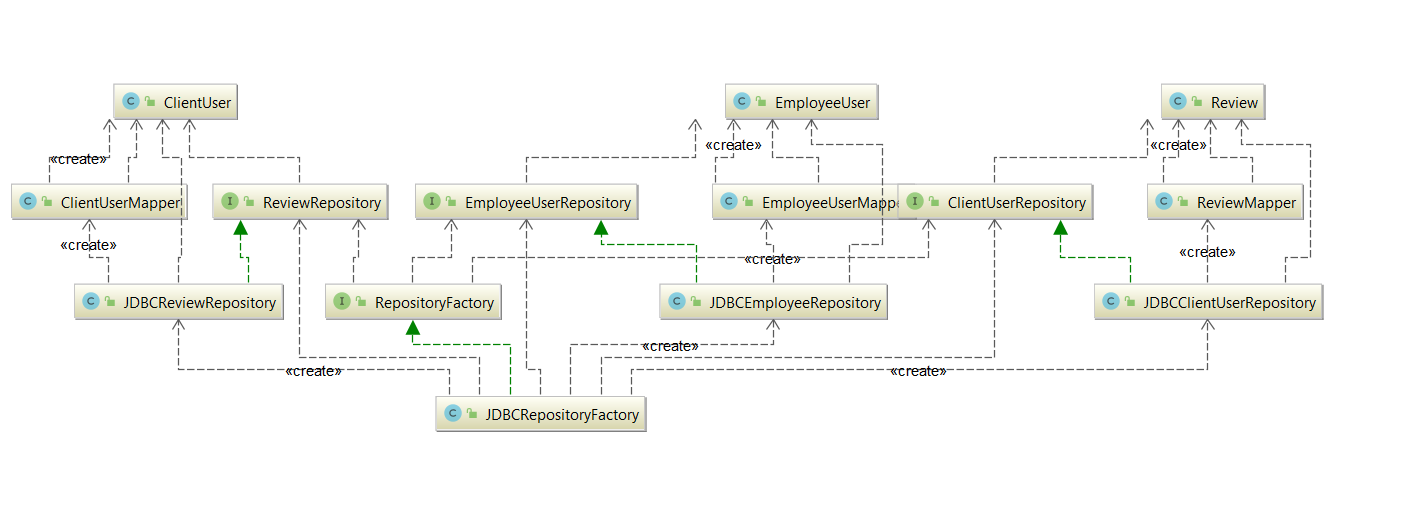


## Class Design

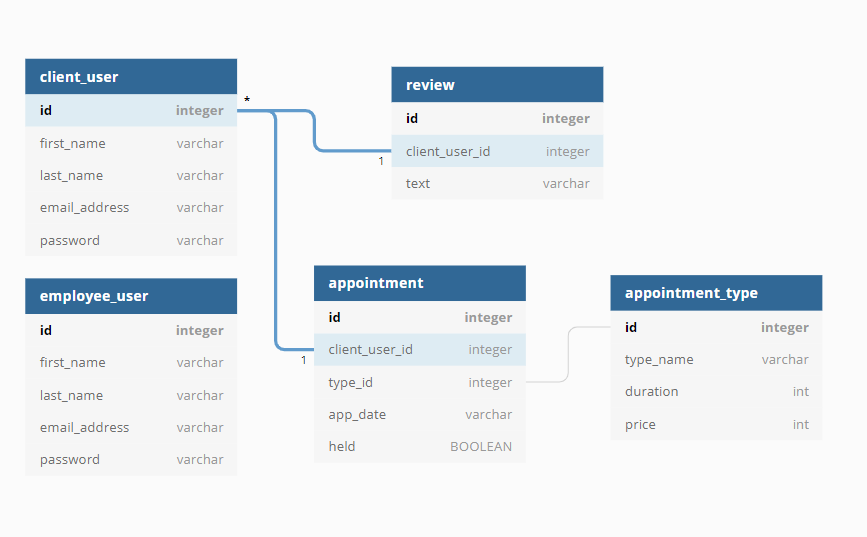
A pattern that I chose to use is the Repository Pattern because it provides a useful way off accessing data in the Business Layer. The business layer only knows what objects it needs and asks for them and the Repository has the responsibility of delivering the objects.

MVC Pattern will be use in the front end.  This pattern is used to separate application's concerns.



**

# Data Model



# Test Strategy

In order to verify the validity of the results, the following tests will be made:

1. Login test (For client and employee)
2. Make an appointment (client)
3. Update the date (client)
4. Leave a review (client)
5. Edit review (client)
6. Confirm an appointment (employee)
7. Delete a review from a user (employee)
8. DataBase test-verify if the information introduced is actually stored.

I will introduce different inputs such that it will be covered all possible cases.

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*