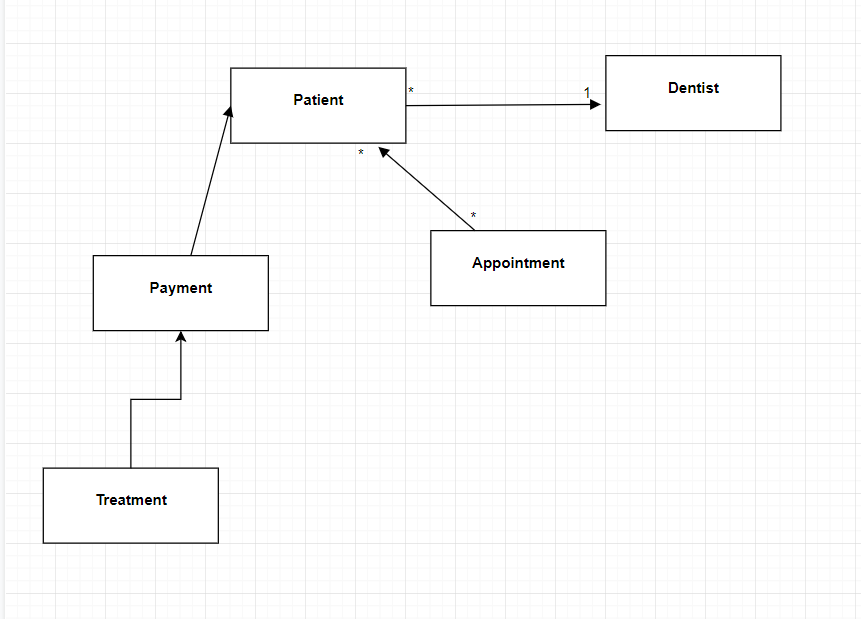
<Dental Clinic Management System>

Analysis and Design Document

# Elaboration – Iteration 1.1

# Domain Model

A domain model is a system of abstractions that describes selected aspects of a sphere of knowledge, influence or activity. The model can then be used to solve problems related tot that domain. The domain model is a representation of meaningful real-world concepts pertinent to the domain that need to be modelled in software. The concepts include the data involved in the business and rules the business uses in relation to that data.



# Architectural Design

## Conceptual Architecture

The architecture that I am going to be using for this web application project is the Layered Architecture. A Layered Architecture is a way of organizing the code such that there are four main categories: presentation layer, business layer, persistence layer and database layer.

* **Presentation layer** contains classes responsible for the User Interface
* **Business Layer** contains rules that determine how data can be created, stored and changed
* **Persistence Layer** deals with persisting(storing and retrieving ) from a database
* **Database Layer** provides access to data stored in database

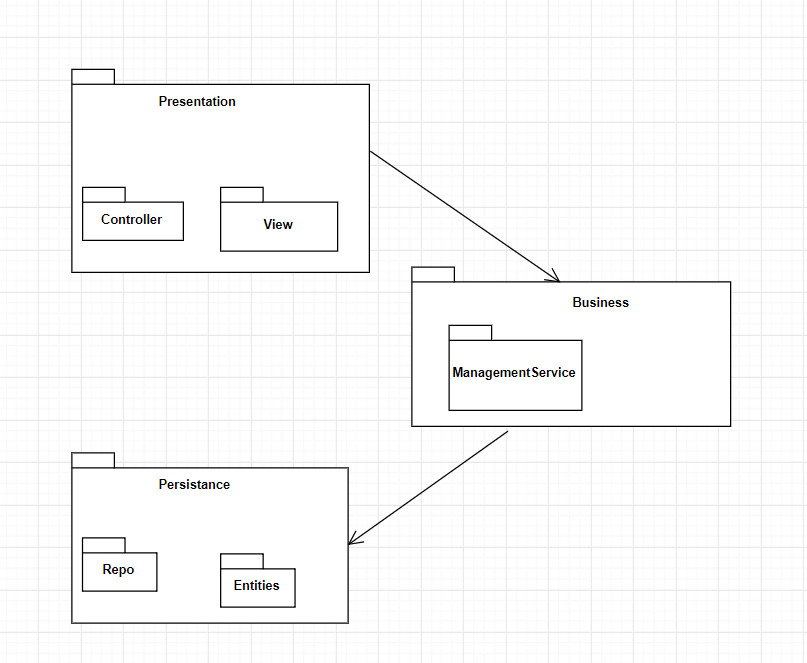
**Benefits of layered architecture:**

* **Simplicity**
* **Consistent across different projects**
* **Separation of concerns**

The design pattern that I am going to use is the MVC pattern: Model, View and Controller. There are many advantages for using this pattern:

* Rapid application development
* While changing the View, the Controller layer doesn’t even have to know where the data goes
* Low coupling between different components(View doesn’t have dependency on service layer and only Controller depends on it)

## Package Design



## Component and Deployment Diagrams

# Component diagram

# 

# Deployment diagram

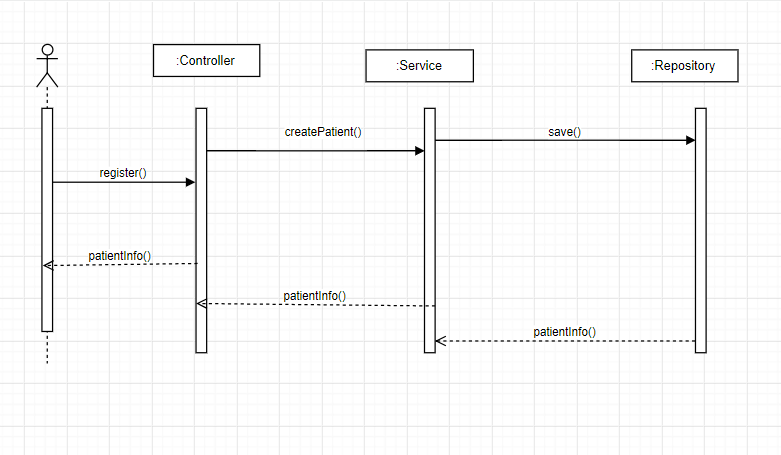
# 

# Elaboration – Iteration 1.2

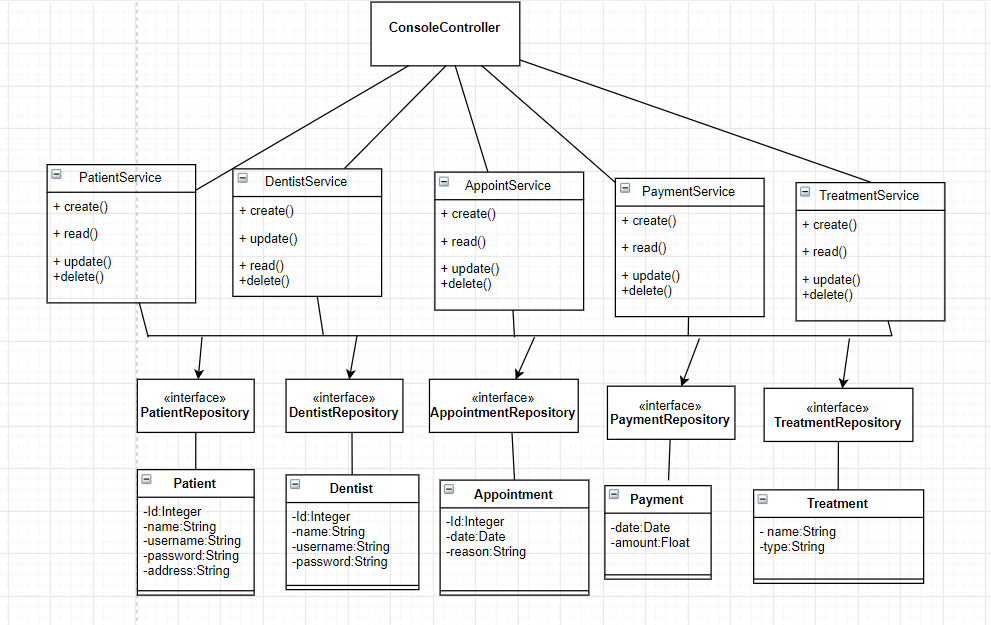
# Design Model

## Dynamic Behavior

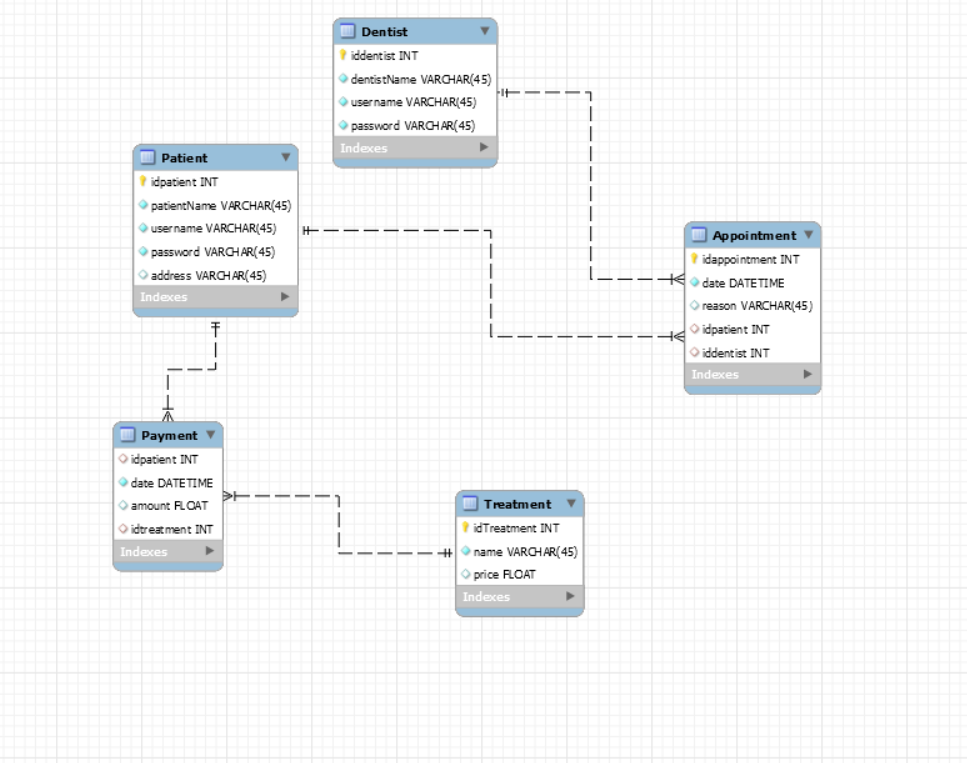
**Create Patient scenario**



## Class Design



# Data Model



# Test Strategy

**Unit testing**

Unit tests are very low level, close tot the source of your application. They consist in testing individual methods and functions of the classes, components or modules used by the application.

**GUI testing**

Graphical User Interface testing is defined as the process of testing the system’s GUI of the application. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars: menu bar, dialog boxes, windows, etc.

# 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]*