Student: Pop David

**Group: 30235**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 4

3. System Architectural Design 6

4. UML Sequence Diagrams 7

5. Class Design 8

6. Data Model 10

7. System Testing 11

8. Bibliography 11

1. Requirements Analysis

# Assignment Specification

Design an application for a tourism agency that has two types of users. Normal users are represented by agents and perform operations on clients and their reservations, and administrators perform operations on agents’ data.

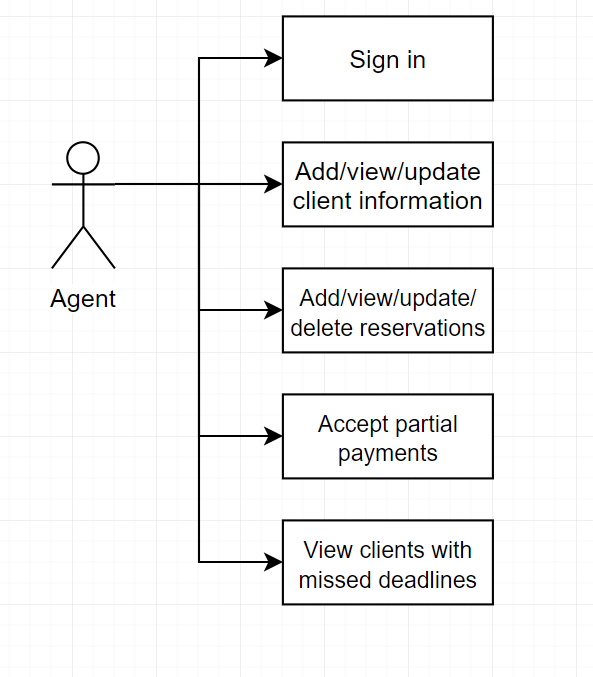
# Functional Requirements

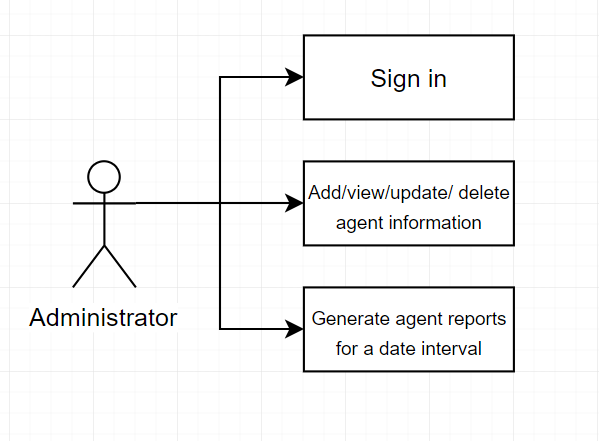
* All input data types are constrained by their respective container
* Details of the clients registered in a reservation are stored one-client-per-line
* After sign in, the application will redirect the flow to the corresponding interface (normal and administrative)
* Entity data is automatically retrieved when the user selects the respective entry in lists
* The application must not be accessible by unauthorized users

# Non-functional Requirements

* The application should be intuitive enough to be properly used knowing only the manual
* The application should not take longer than 0.5 seconds to respond to a request

2. Use-Case Model





**Use case**: Update reservation

**Level**: user goal

**Primary actor**: agent

**Main success scenario**: The agent selects a client in the client list, and then selects a reservation in the reservation list. After making changes to the data, the agent clicks the “Save” button.

**Extensions**: None.

3. System Architectural Design

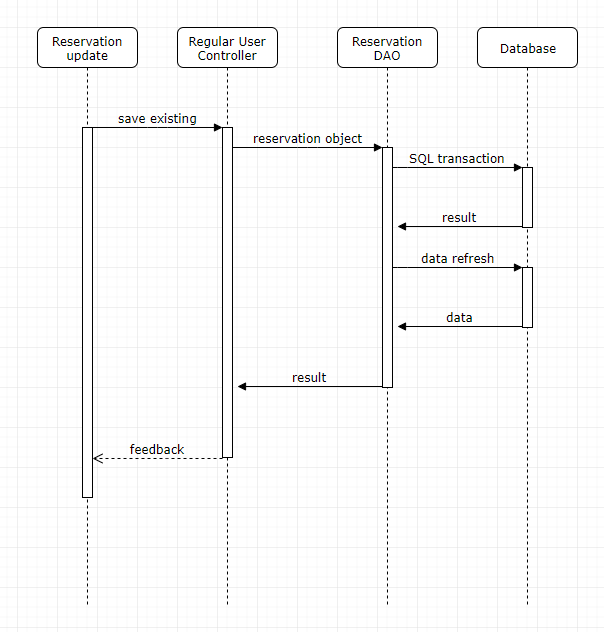
**3.1 Architectural Pattern Description**

The main architectural pattern used in this project is the Layers pattern. The layers implemented are: presentation layer, business layer, persistence with databases. The order described also represents the dependency of one layer: the business layer does not depend on the presentation layer, and one layer cannot skip past another layer downstream.

**3.2 Diagrams**

[](https://www.safaribooksonline.com/library/view/software-architecture-patterns/9781491971437/assets/sapr_0101.png)

4. UML Sequence Diagrams



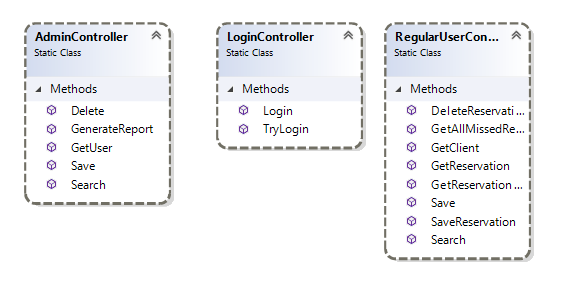
5. Class Design

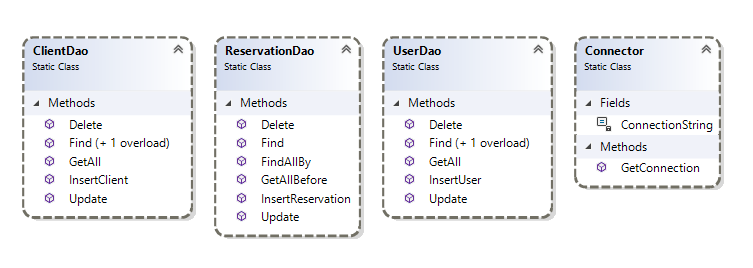
**5.1 Design Patterns Description**

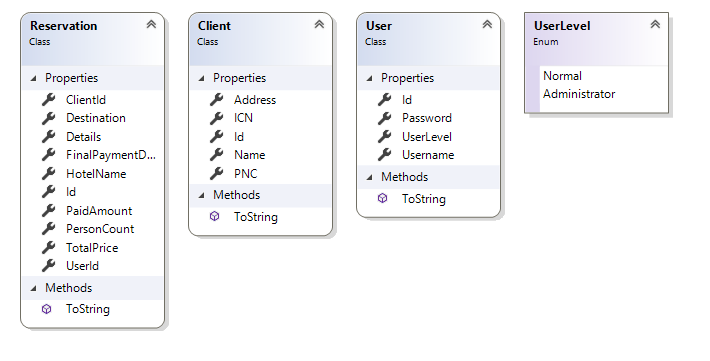
The domain model pattern is used as a domain logic pattern. Database tables can be mapped to classes.

The table data gateway pattern is used as a data source pattern. The DAO classes contain all the relevant database abstraction.

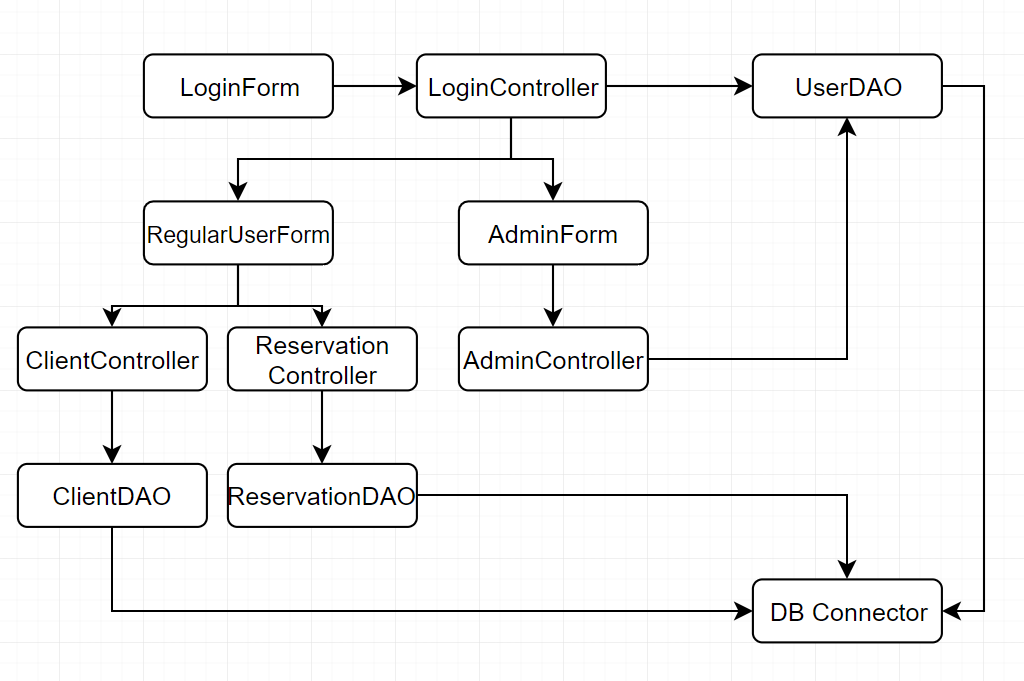
**5.2 UML Class Diagram**

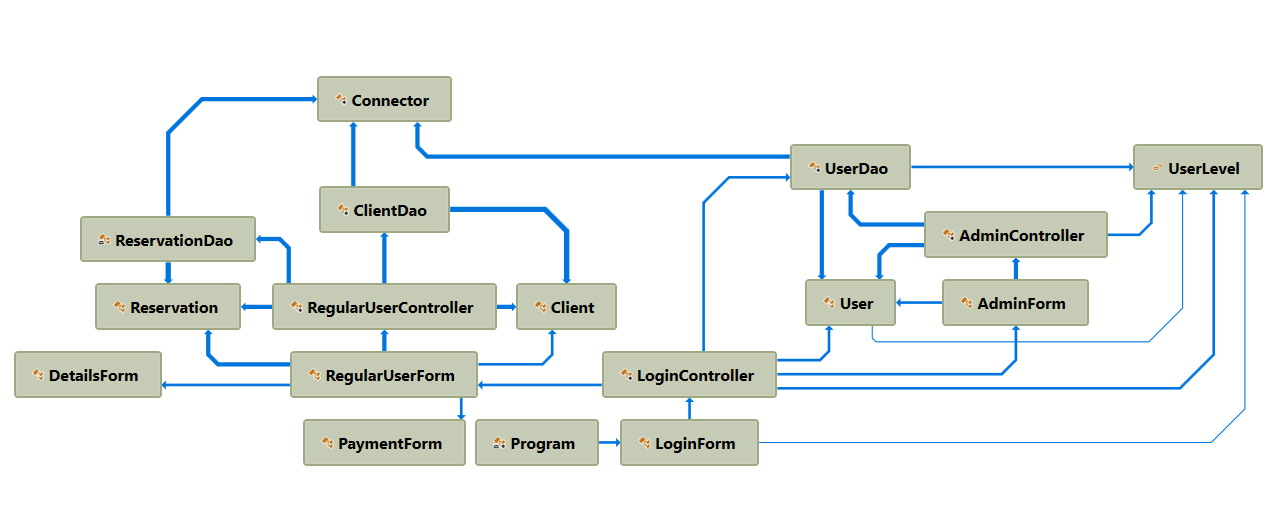


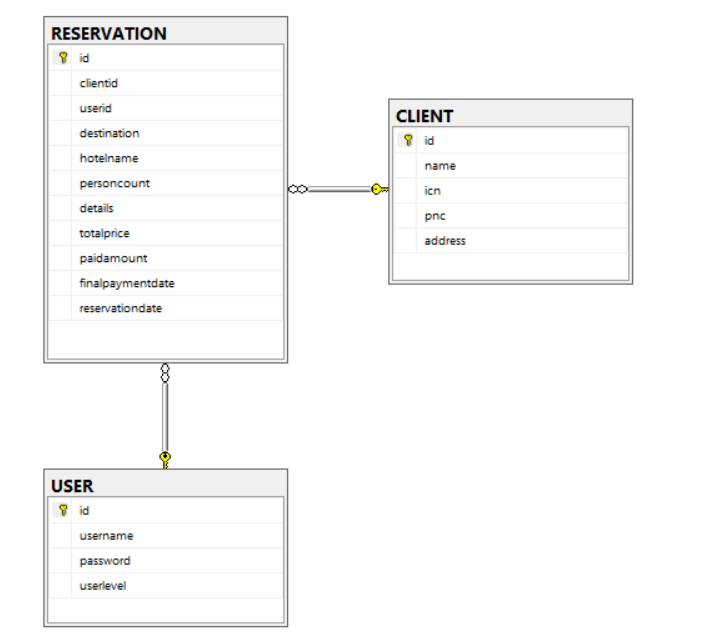




6. Data Model







7. System Testing

Testing has been done manually after each operation.

8. Bibliography

Martin Fowler et. al, Patterns of Enterprise Application Architecture, Addison Wesley, 2003

<http://docs.oracle.com/javase/tutorial/uiswing/>

<http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html>

<https://msdn.microsoft.com/en-us/library/54xbah2z(v=vs.110).aspx>

<https://msdn.microsoft.com/en-us/library/e80y5yhx(v=vs.110).aspx>