Art Gallery Management

Analysis and Design Document

Student:Pop Laura-Maria

**Group:30233**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <05/APR/2017> | <1.0> | - | Pop Laura-Maria |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

*AGM will be an online Art Gallery Management application. It will have two types of user:*

1. *Regular user*
2. *Admin*

*Through a Data Base, the application will keep information about:*

1. *Artists:*
   1. *Name*
   2. *Birthplace*
   3. *Age*
   4. *Style of art*
2. *Art Work:*
   1. *Artist*
   2. *Year it was made*
   3. *Unique title*
   4. *Type of art*
   5. *Price*
3. *Type of Art:*
   1. *It can be various (like renaissance or work of the 19th century, etc.)*
4. *Customers:*
   1. *Name*
   2. *Address*
   3. *Email*
   4. *Phone*
   5. *Maybe a short description for their profile, but it is not obligatory*
   6. *Likes of arts*
   7. *Total amount they spent on Gallery*
5. *Administrators:*
   1. *Name*
   2. *Address*
   3. *Email*
   4. *Phone*
   5. *Etc.*

*The regular user can perform the following operations:*

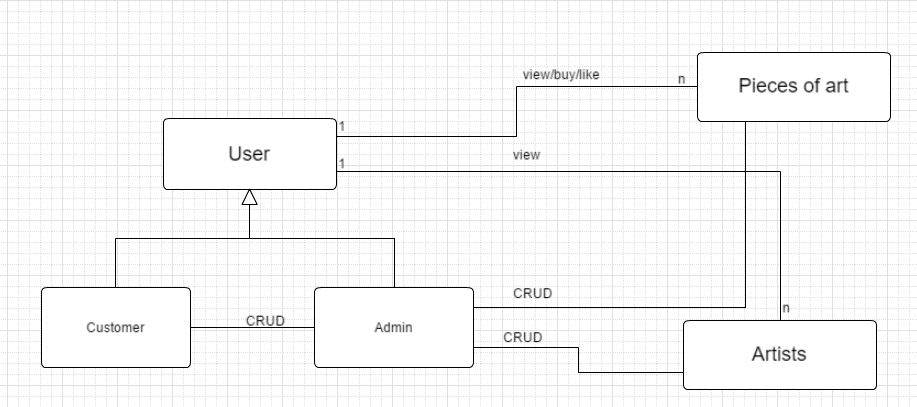
1. *To like a piece of art*
2. *View art*
3. *Buy art*
4. *Register on the site*
5. *Log-in/Log-out*

*The Admin can perform the following operations:*

1. *All operation that a customer can do except Register*
2. *CRUD on artists*
3. *CRUD on customers*
4. *CRUD on pieces of art*
5. *Generate reports*

# Elaboration – Iteration 1.1

# Domain Model

**

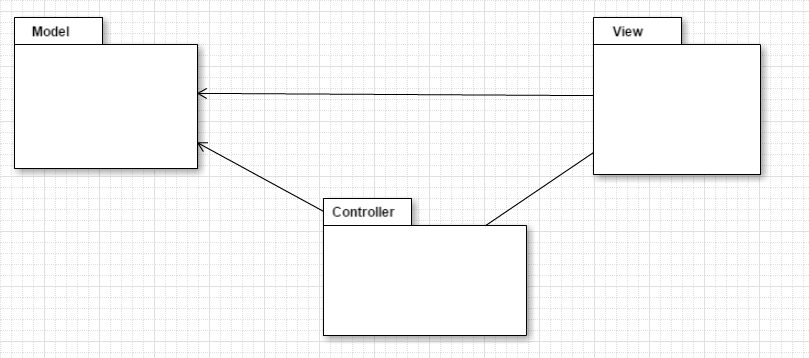
# Architectural Design

## Conceptual Architecture

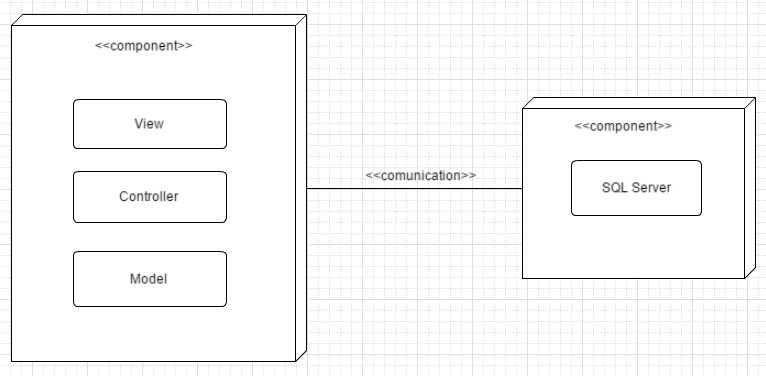
*I will use a MVC (Model-View-Controller) because is perfect considering that the application will be online. A lot of frameworks work with this MVC pattern.*

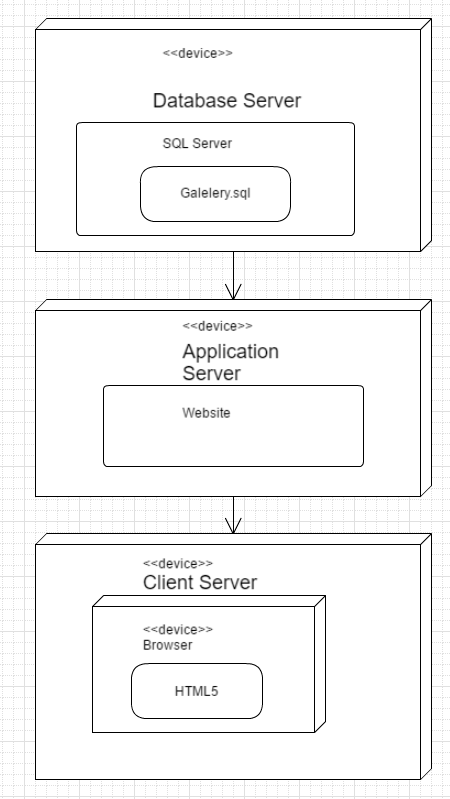
*It will be easy to implement a MVC pattern and it will be useful because the application operations had to be very clear.*

## Package Design

**

## Component and Deployment Diagrams

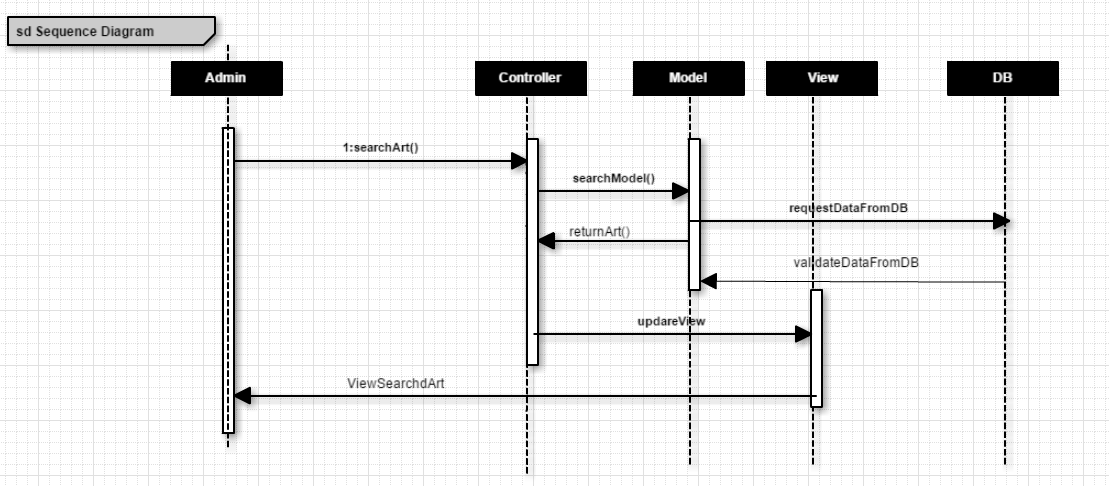




# Elaboration – Iteration 1.2

# Design Model

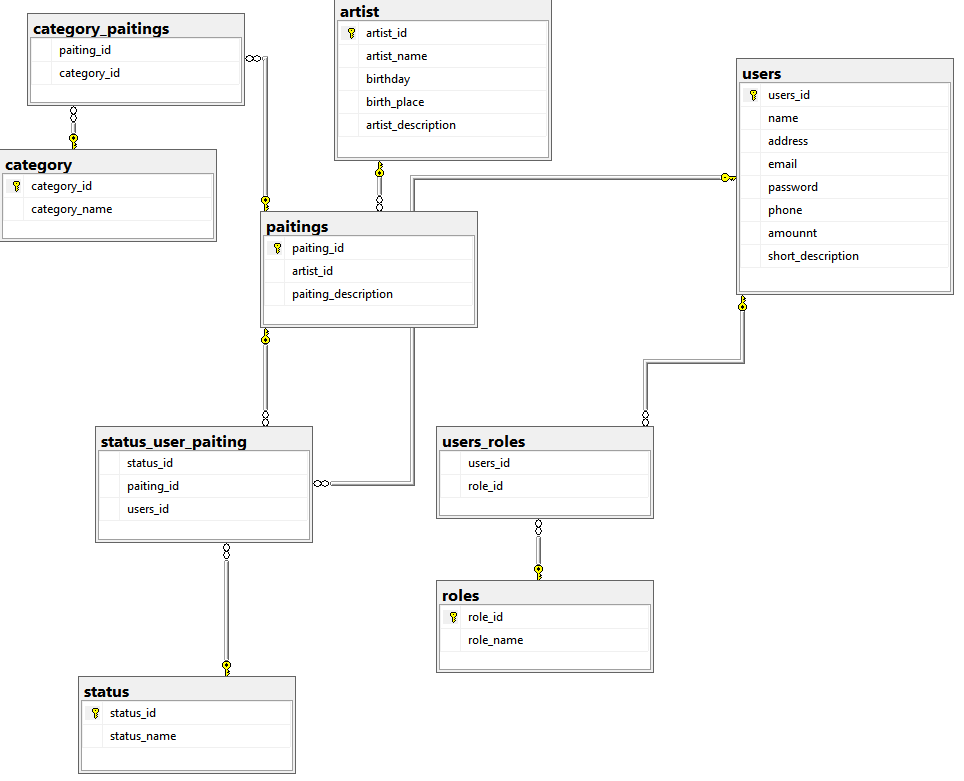
## Dynamic Behavior



## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

**

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# 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]*

# Bibliography