Art Museum Application (AMA)

Analysis and Design Document

Student: Rednic Ana

**Group: 30432**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 19/04/2018 | 1.0 | Domain Model, Architectural Design, Deployment Diagram | Rednic Ana |
| 25/04/2018 | 1.1 | -added- Design Model, Data Model | Rednic Ana |
| 17/05/2018 | 1.2 | -revised- Diagrams | Rednic Ana |
| 20/05/2018 | 1.3 | -revised- Diagrams and text (Final form) | Rednic Ana |

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 Deployment Diagram 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

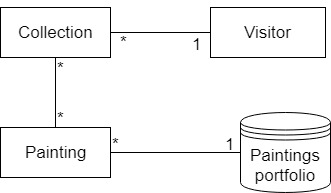
# Project Specification

The Art Museum Application (AMA) will be a stand-alone desktop application that will provide a way for the visitors of the Art Museum of Baia Mare to easily find information about the works of art in the museum. The AMA could also be a support for the ones that cannot visit the museum in person: they can do a virtual tour using the application.

# Elaboration – Iteration 1.1

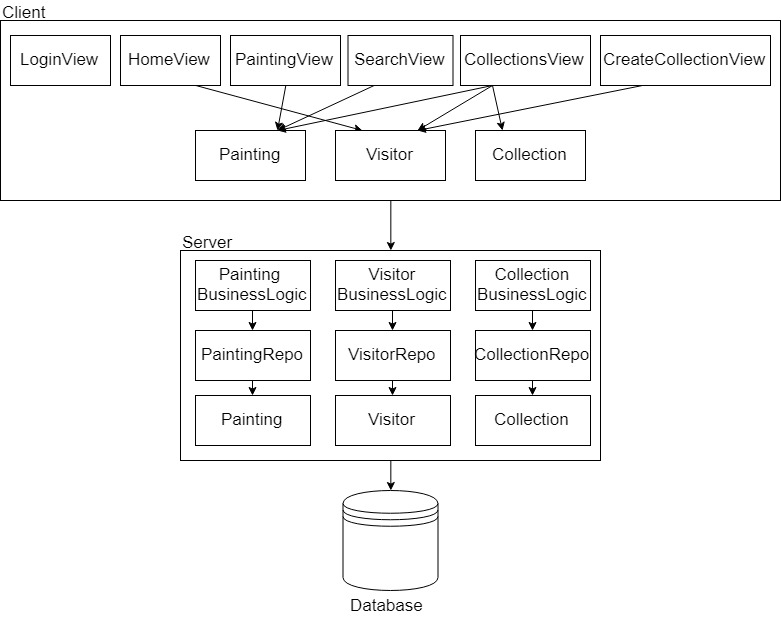
# Domain Model

The domain model is a conceptual model of the domain that incorporates both behavior and data.



# Architectural Design

## Conceptual Architecture

I chose to implement the 3 Layers Architecture and Client-Server architecture. The 3Layers are the Presentation Layer, the Business Logic Layer, and the Data Layer, arranged the following: Data Access Layer and Business Logic Layer are implemented in the Server. The Presentation Layer is implemented in the Client. In the Data Access Layer, I make the connection to the database. In the Business Logic Layer I process the information. The Presentation Layer is the Client user interface.

## Package Design

## C:\Users\Ana\Downloads\Package Diagram project (1).jpg

## Deployment Diagram

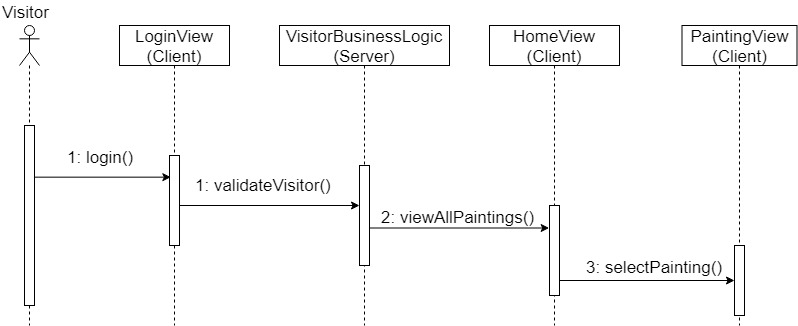
# C:\Users\Ana\Downloads\Deployment Diagram (3).jpg

# Elaboration – Iteration 1.2

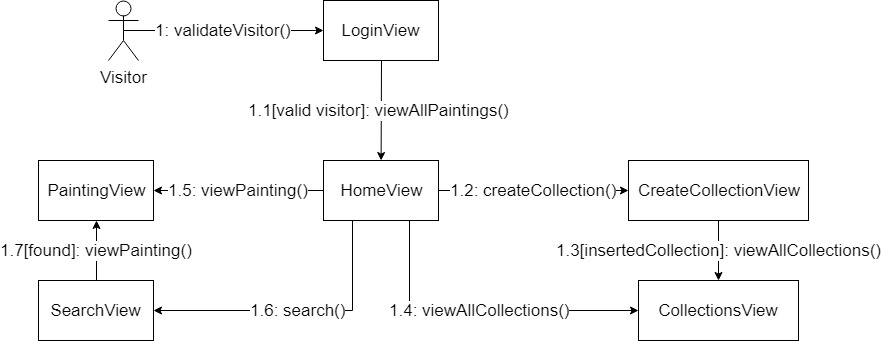
# Design Model

## Dynamic Behavior

**Sequence Diagram**



**Communication Diagram**



## Class Design

I implemented three design patterns: Observer, Chain of Responsibility, and Interpreter. The Observer is used to notify the visitor when a new Collection has been created and to update his list of collections. The Chain of Responsibility is used to select the command sent from the Client to the Server. The server takes the command and goes with it through the whole chain until it matches one command from the chain and executes the related responsibility. The Interpreter is used to interpret the words search by the visitor in the Search function. Depending on the search input, the visitor receives from the interpreter, besides the list of correlated Paintings, an interesting fact about his search.

# C:\Users\Ana\Downloads\Class diagram project (1) (1).jpg

# Data Model

There are 3 main tables: Painting, Collection, and Visitor. There is a One-to-Many relationship between Visitor and Collection, and a Many-to-Many relationship between Painting and Collection.

# C:\Users\Ana\Downloads\Data Model project (1).jpg

# Unit Testing

Unit testing will be performed on each component of the system, validating it. The focus falls on the tests that impact the behavior of the system. This will be done using Dataflow testing.

In the Integration testing part, individual units will be combined and tested as a group. This will look forward to expose defects in the interfaces and in the interaction between integrated components.

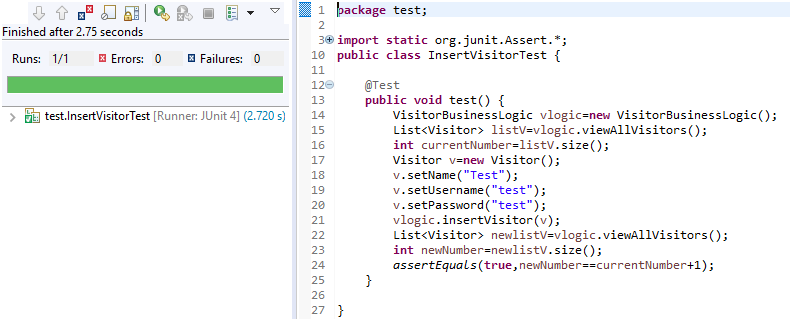
The System testing part will verify if the whole application meets the specified requirements.

Validation testing will be performed to decide whether or not the application is ready to be made available to the end-users. The last three stages will be done using Black Box testing.

# Construction and Transition

# System Testing

In order to test a flow of the application, I created a JUnit test case that checks if a Visitor is correctly inserted in the database. It computes the current number of visitors in the database, inserts a new one, computes again the number of visitors, and the new number should be equals to the old one plus 1.



# Future improvements

As a future improvement, I would consider the possibility of an administrator to enter the application and insert or modify paintings. Also, I would make the interpreter process and give result to any kind of data searched by the user. The paintings’ descriptions should be more complex. Another function to add to the application is the scan QR part: the user could search for a painting by scanning a QR code with the device’s camera. Nevertheless, the application should be developed on Android devices.

# Bibliography

* <https://en.wikipedia.org/wiki/Domain_model>
* <http://softwaretestingfundamentals.com/>
* <https://www.tutorialspoint.com/software_testing_dictionary/data_flow_testing.htm>
* <https://www.tutorialspoint.com/design_pattern/interpreter_pattern.htm>
* <http://www.blackwasp.co.uk/ChainOfResponsibility.aspx>