<Numismatic Application>

Analysis and Design Document

Student:Urda Sebastian George

**Group:30433**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <1/Apr/18> | <1.0> | <Update Iteration 1.1> | <Urda Sebastian George> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

The purpose of this application will be to serve as my final project at Software Design. The main goal of the Numismatic Application App will be to create a common platform for the collectors who want to share the information they know about the coins they own. The application will give the coin collectors the opportunity to keep track of their coins and communicate with others who share the same passion. A user could create an account, log in, add coins, se the coins that another user shared.

# Elaboration – Iteration 1.1

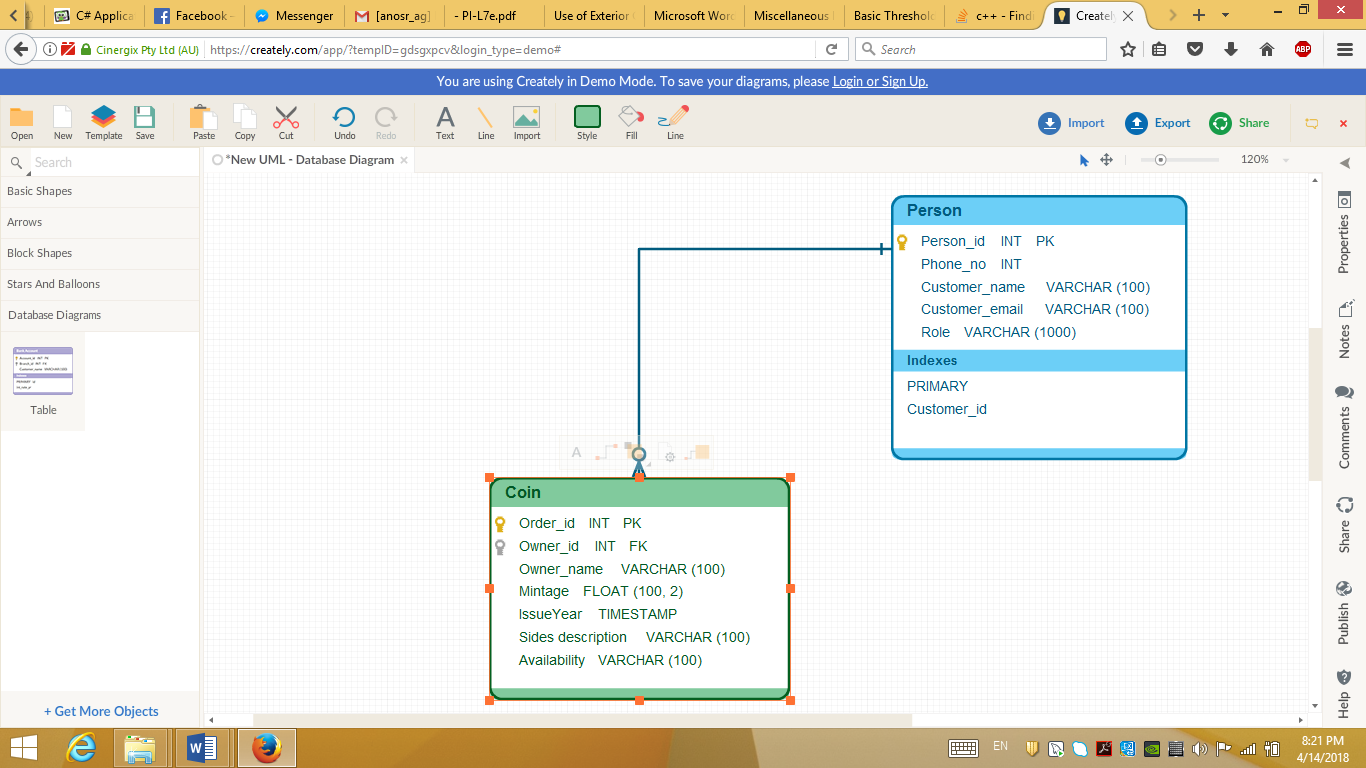
# Domain Model

The database will consist of at least 2 tables: Person and Coin.

The Person table will also contain the role of the person: collector, historian, seller .

The historian will also act like an admin user.

The Coin table will have the specifications of the Coin: the year it was issued, mintage, alloy and 2 pictures one for each side, the number of coins the collector owns and weather is available for sale or not.



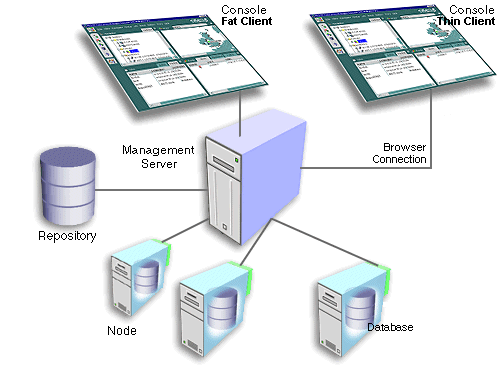
# Architectural Design

## Conceptual Architecture

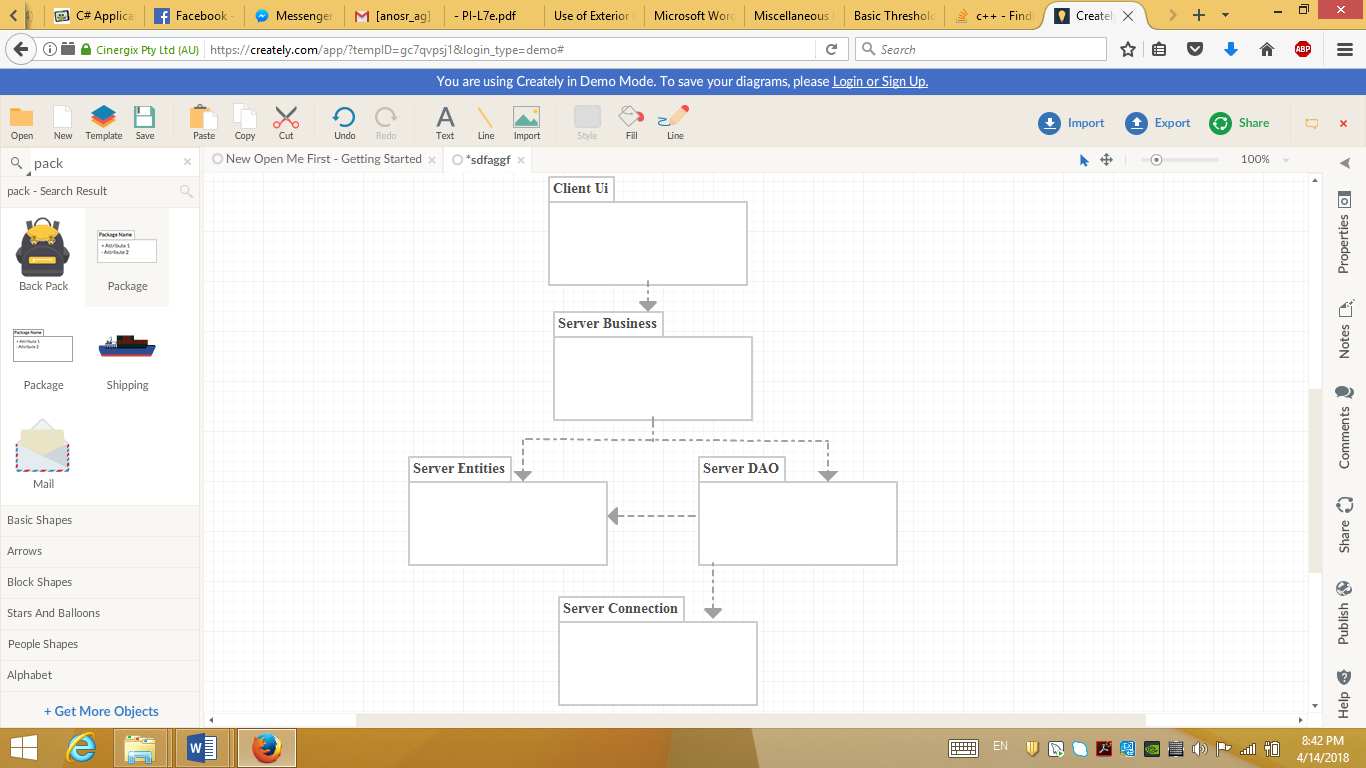
The system uses the client-server architecture. It is an obvious choice because of the need of many users and one processing unit which needs to provide services for each of these users.

* The server will take care of accessing the database, as well as perform the business logic for the user commands. Therefore, it will be a Fat Server.
* The client will display the user interface and send commands to the server via the Internet connection. Therefore, it will be a Thin Client.

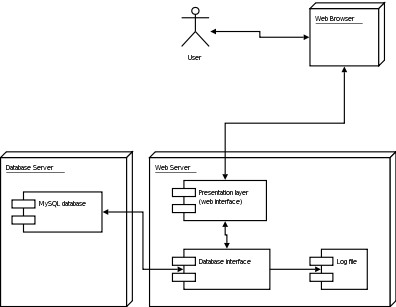
This type of approach is called **Sever-based processing**, since the server does the hard part while the client only displays data to the user and calls the server for any type of transaction that needs to be performed.



## Package Design



## Component and Deployment Diagrams



# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

## Class Design

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

# Data Model

*[Create the data model for the system.]*

# 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