Art Collection and Discovery

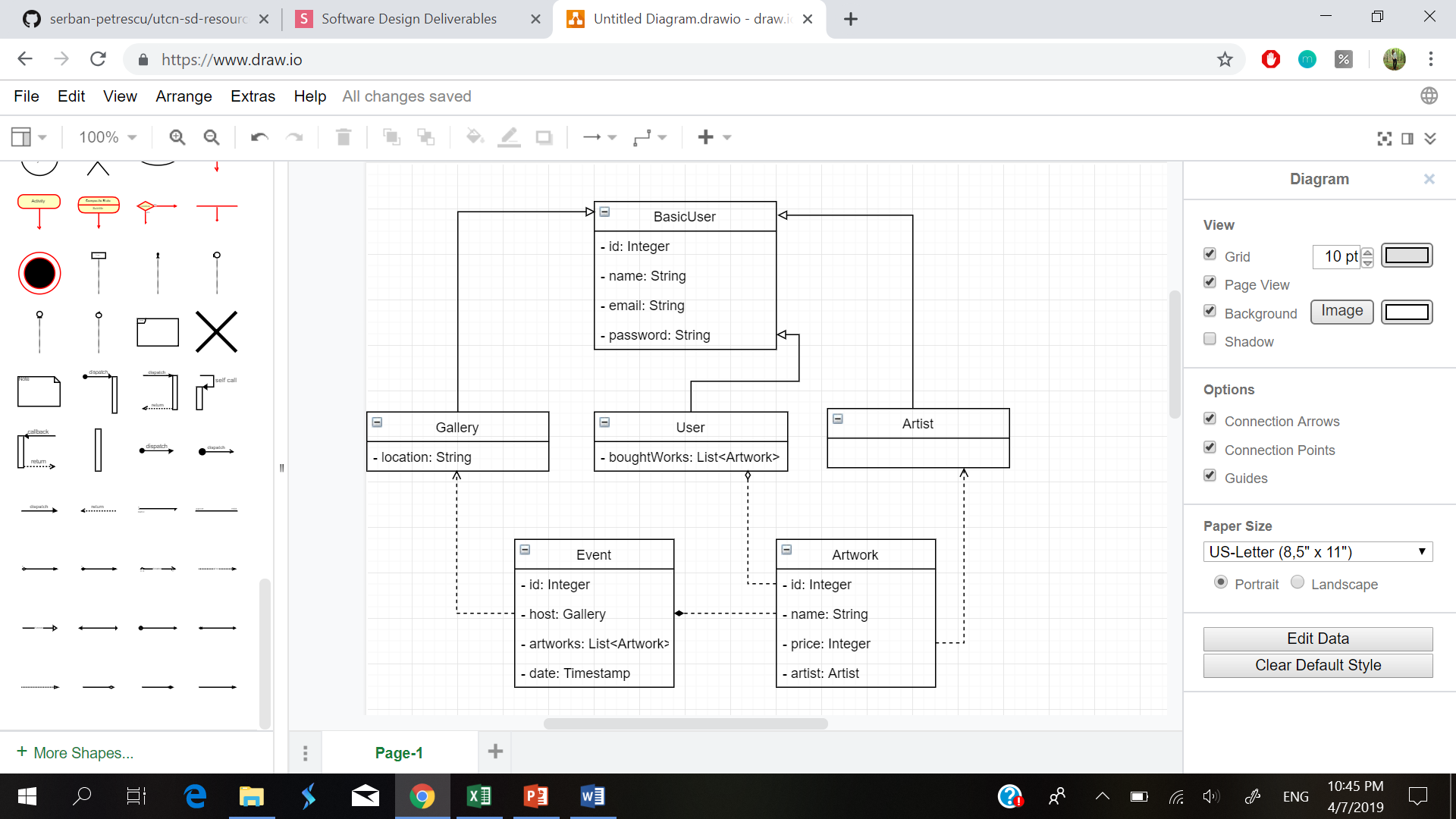
Analysis and Design Document

# Elaboration – Iteration 1.1

# Domain Model

Since the app is designed for artists, galleries, as well as art enthusiasts, the following entities must be present:

* Artist
* Artwork
* User
* Gallery
* Event

Since the artist, the buyer, and the gallery all have names and credentials, they can inherit those features from a basic user, and add their specific fields (and later have specific methods associated to each one of them).

# Architectural Design

## Conceptual Architecture

The architecture is going to be multi-layered. It is a simple architecture, which we have worked with before, easy to understand and use, and it offers a clear separation based on roles: each layer handles different types of functionalities. The presentation layer contains the UI components, while the model includes the aforementioned classes. The service layer handles all operations that can be performed on and by the entities in the model. The DAOs, as their name suggests, are objects which access and retrieve the data, one object corresponding to each entity. Last but not least, there is a database, designed in MySQL, which will contain the tables with all the data needed by the application.

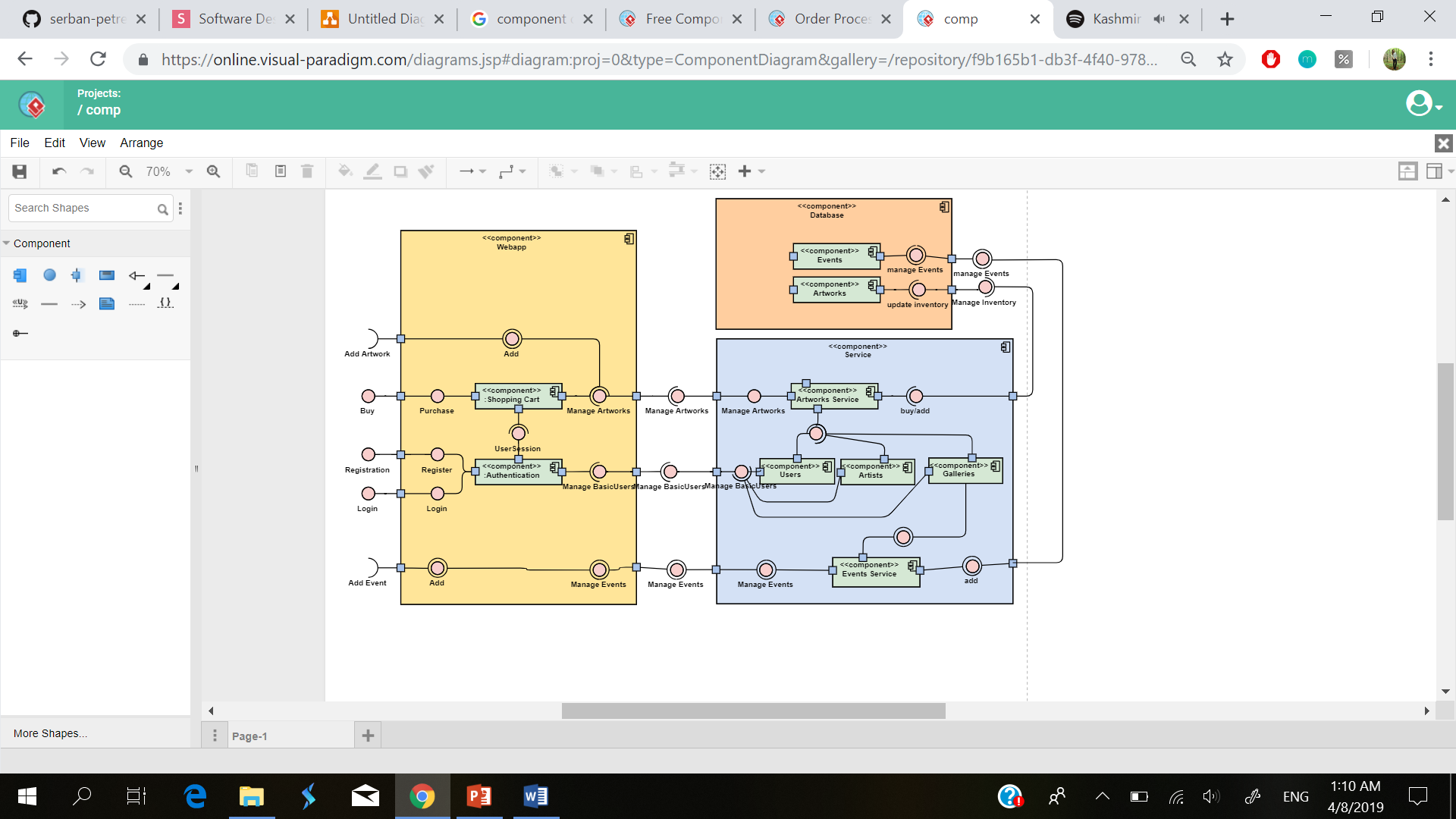
## Package Design

## 

## Component and Deployment Diagrams

# Deployment Diagram:

# 

 Component Diagram:

# 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.]*

# Test Strategy

*[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]*