# Bike portal
# Analysis and Design Document
**Student: Ács Dávid**
**Group: 30432**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 18/Mar/18 | 0.1 | Housekeeping | Ács Dávid |
| 04/Apr/18 | 0.2 | Architecture and views | Ács Dávid |
| | | | |
| | | | |

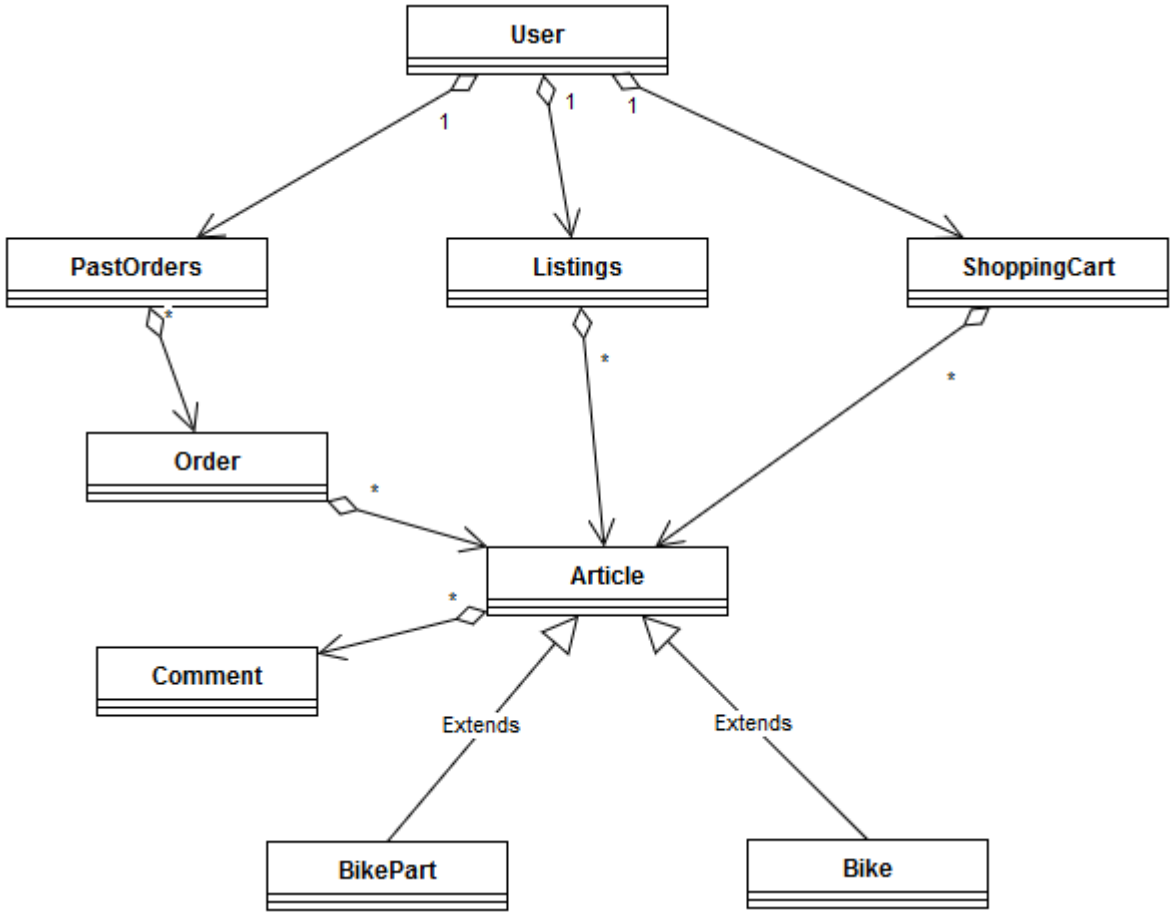
# Table of Contents

# I. Project Specification

Bike portal is an online Bike and bike parts store. User can buy/sell bikes and related parts online. To buy products, the user must be registered and logged in, and should add the products to the shopping cart. To check out the user can pay by credit card. Users may rent bikes, and the admin may approve these requests. The admin of the portal can delete, modify announcements/listing if they are inappropriate.

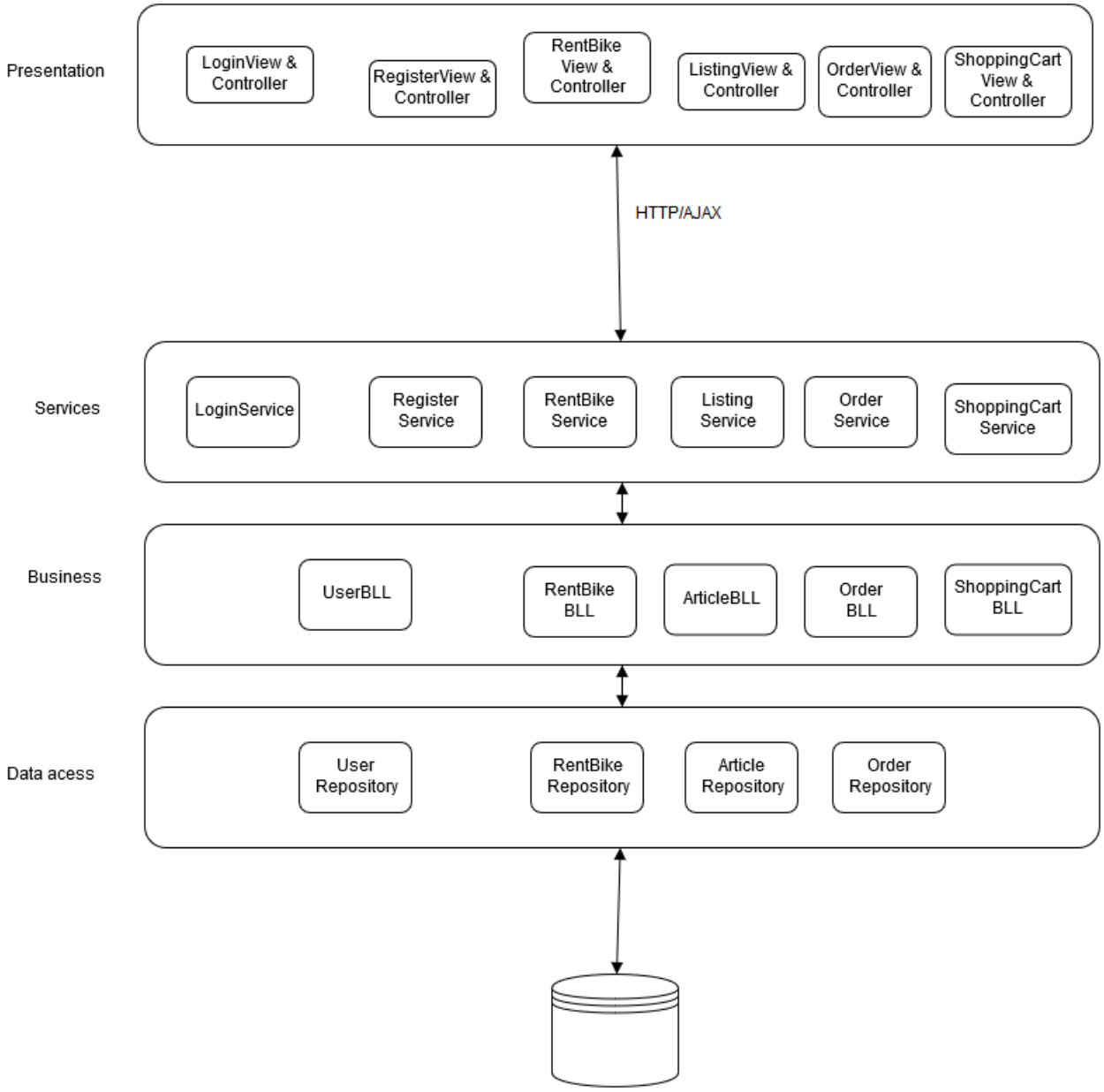
# II. Elaboration – Iteration 1.1

## 1. Domain Model



# 2. Architectural Design

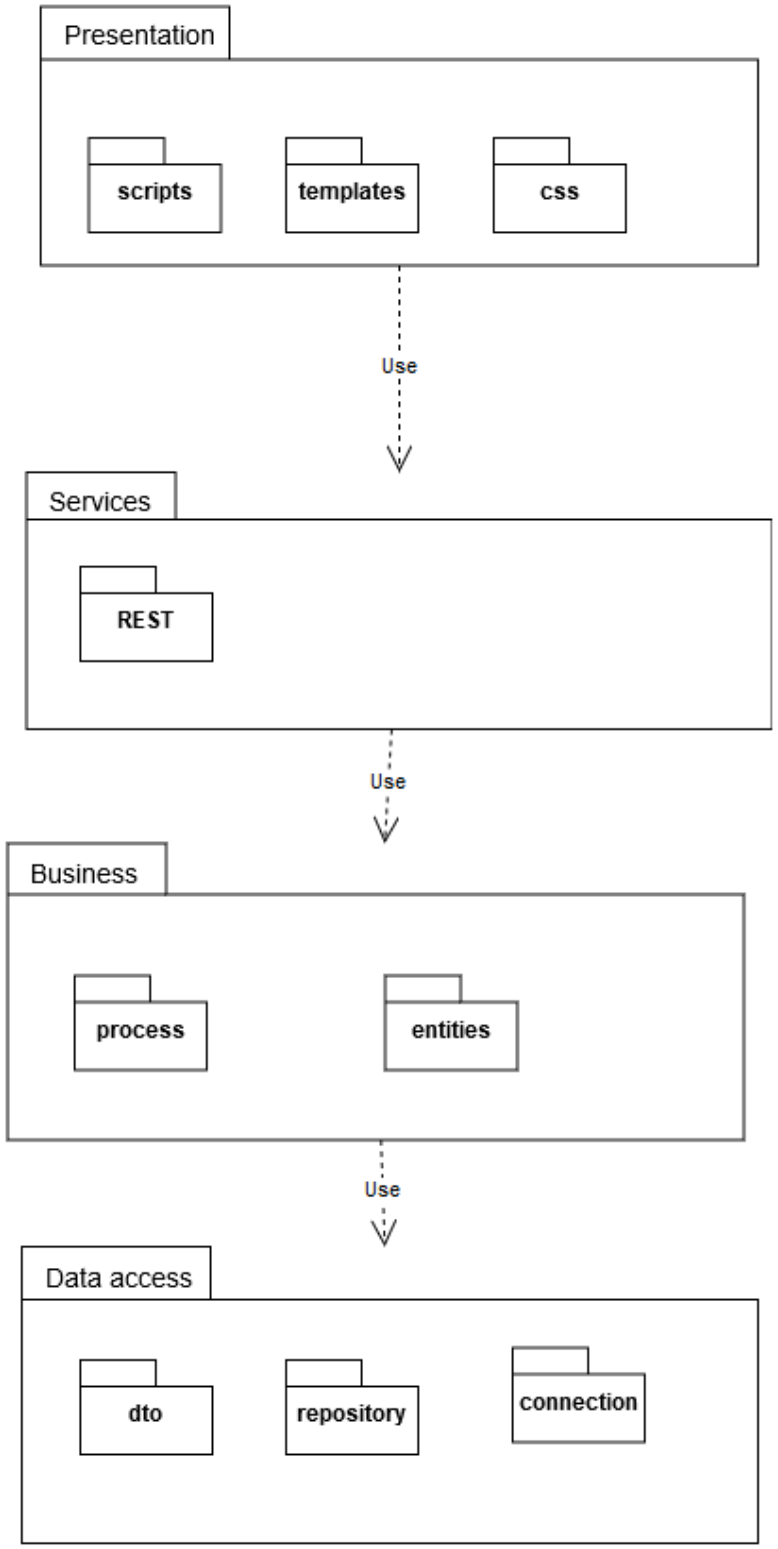
## 2.1 Conceptual Architecture




Client server architecture is used in this architecture, to separate the server-side application (data and logic) from the client-side (UI). This way the client-side and the server side may evolve independently of each other.

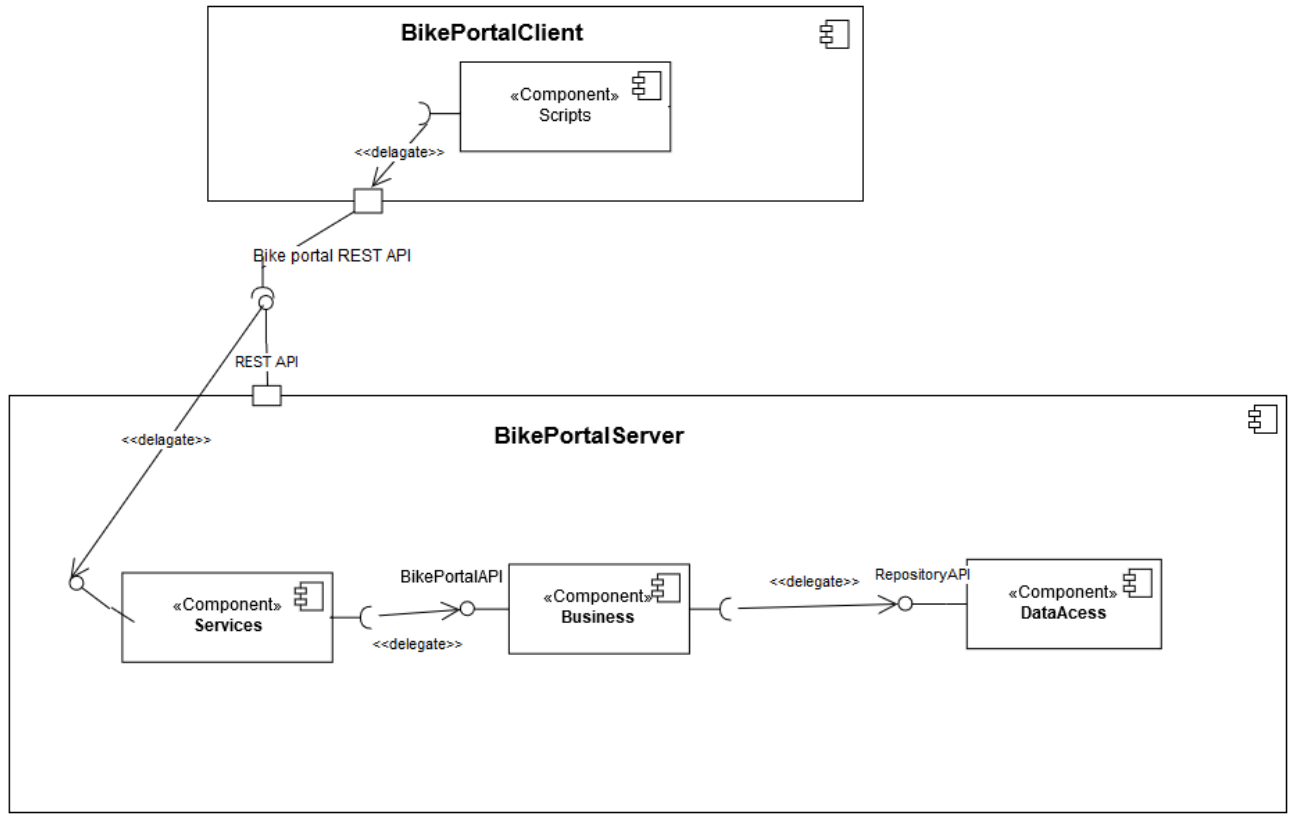
The separation can be observed, where the HTTP/AJAX calls are made over the network.

## 2.2 Package Design

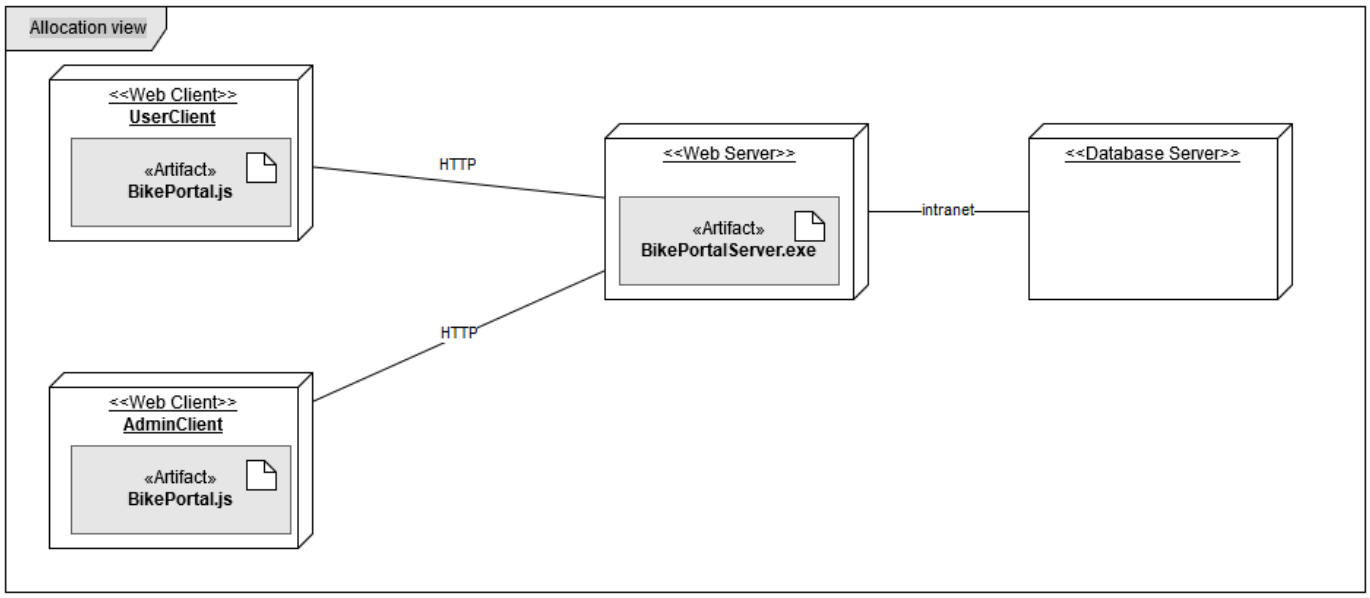



The package diagram of the system represents the layered structure of the application.

### 2.3 Component and Deployment Diagrams




The component diagram shows the clear delimitation between the server and the client. They can only communicate through HTTP, or as it is shown in the diagram, through the REST API.




In the allocation view we may notice that there may be more than one client, and they role may differ depending on who is logged in.

# III. Elaboration – Iteration 1.2

## 1. Design Model

### 1.1 Dynamic Behavior

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

### 1.2 Class Design

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

## 2. Data Model

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

## 3. Unit Testing

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

# IV. Elaboration – Iteration 2

## 1. 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.]*

## 2. Design Model Refinement

*[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# V. Construction and Transition

## 1. System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

## 2. Future improvements

*[Present future improvements for the system]*

# VI. Bibliography