Ecommerce Web Application for Outdoor Activities

Analysis and Design Document

Student: Andrei Baraian

**Group: 30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <25/04/18> | <1.0> |  | Andrei Baraian |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

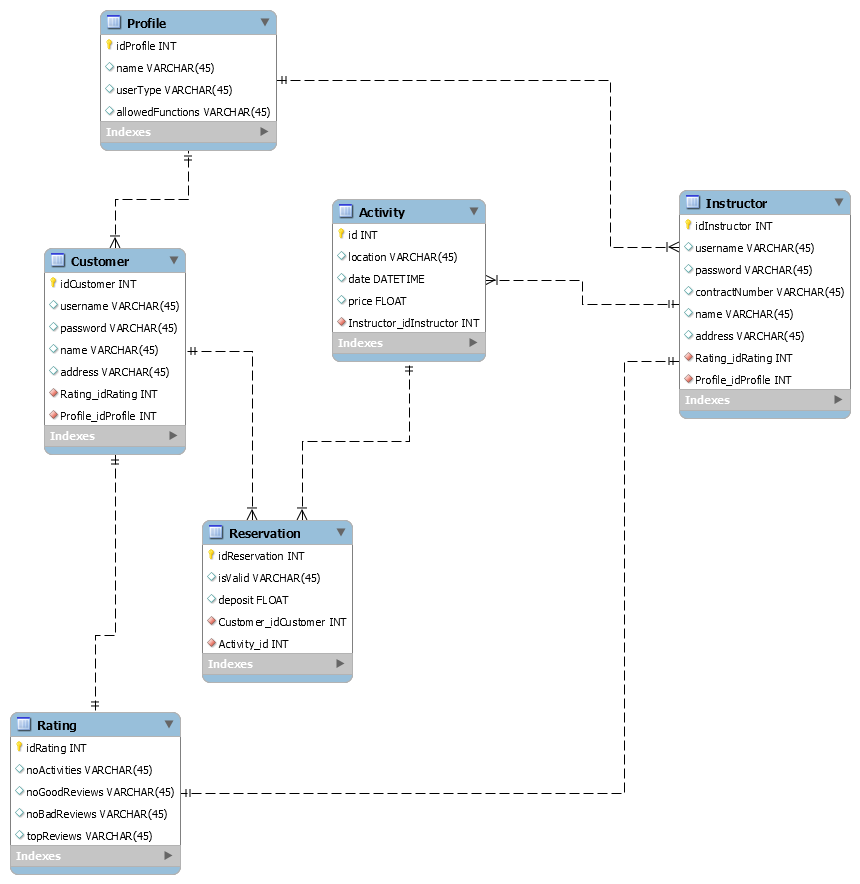
# Project Specification

The project consists of designing an e-commerce platform where outdoor companies can expose their services and products, while clients can buy them and interact with the outdoor instructors from each company. The application will be a web app, which will have different interfaces, depending on the profile of the user.

# Elaboration – Iteration 1.1

# Domain Model

The domain model can be identified by looking at the main actors involved in the system. The following entities have been derived: Customer, Instructor, Profile, Activity, Reservation, Rating.



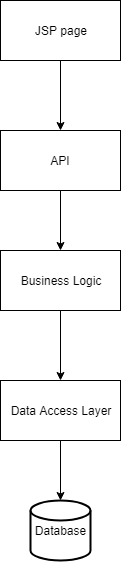
# Architectural Design

## Conceptual Architecture

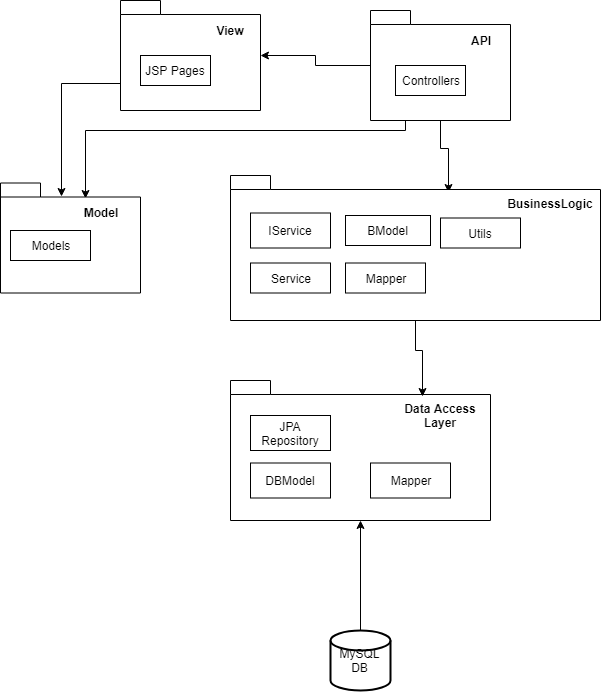
The project is implemented using the Layered Architectural Pattern, along the MVC (Model-View-Controller). The server-side will be implemented in Java and the client one will have JSP pages that will be rendered for the user.

There will be 4 layers: Controller, Business Logic, Data Access and Model. The application will be a RESTful one, following the REST principles. The flow of the interaction with the app will start from the API layer, where a request will be mapped to one of the controllers’ method and then a subsequent call to the business logic layer will be made, to perform the desired operation. The request will propagate until the database, where data will be persisted and the changes (response) will be propagated upwards to be rendered on the user interface.

The layers will have the following relationship:



## 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