<Application for Sports Bookings>

Analysis and Design Document

Student: George Baraian

**Group:30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <17/03/2020> | <1.0> | <details> | <George 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

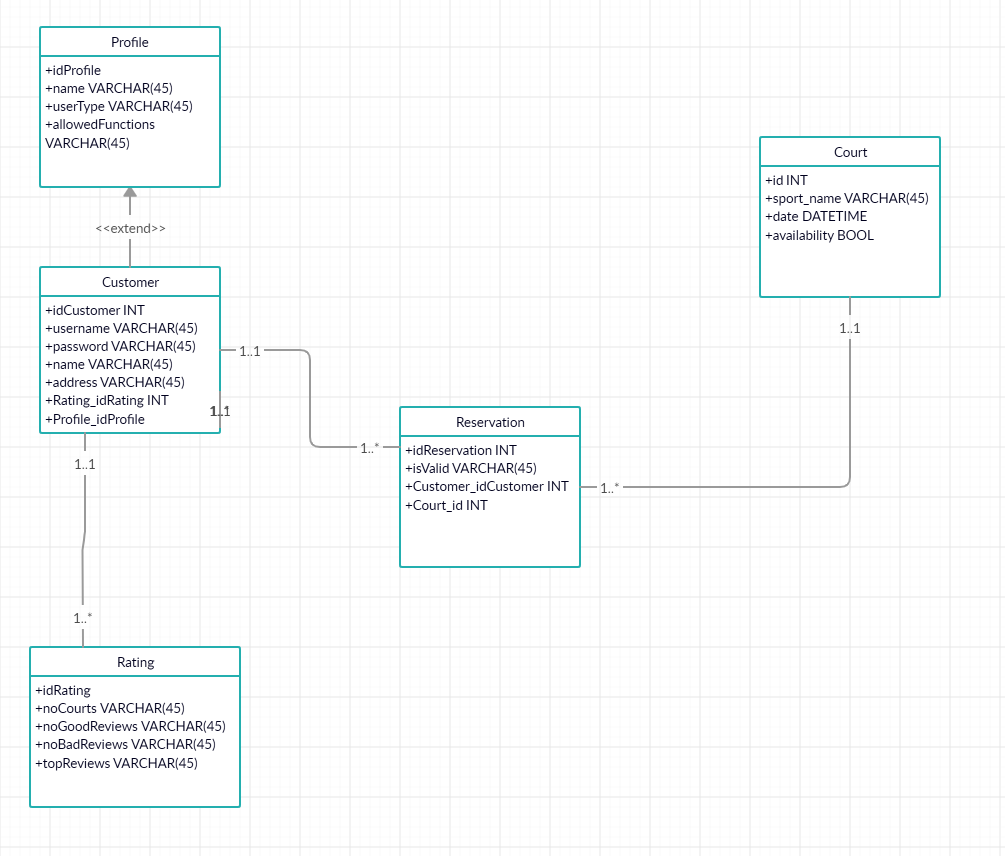
# Project Specification

The project consists of designing of s sports bookings platform where sports facility companies can expose their services, while clients can book them and interact directly with these companies.

# 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, Profile, Activity, Reservation, Rating.



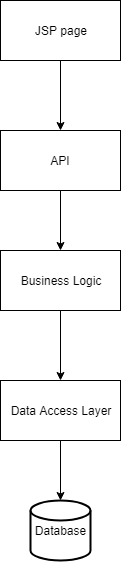
# Architectural Design

## Conceptual Architecture

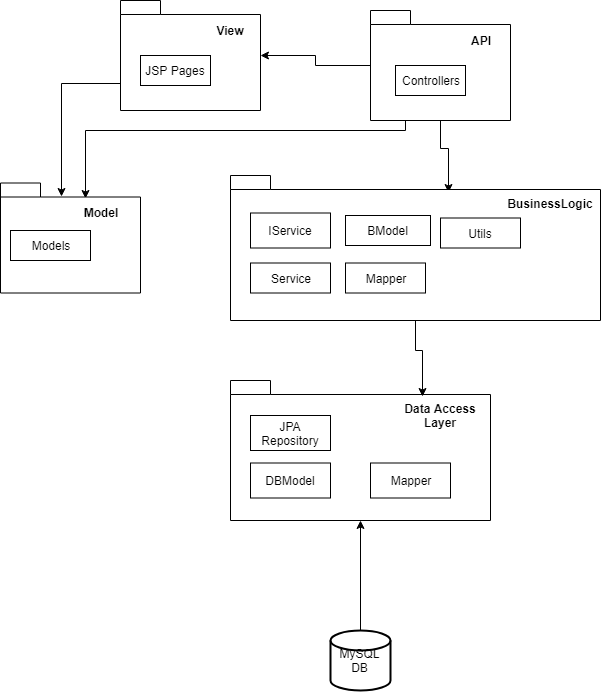
The project is implemented using the Layered Architectural Pattern, along the MVC (Model-View-Controller).

There will be 4 layers: Controller, Business Logic, Data Access and Model. 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.

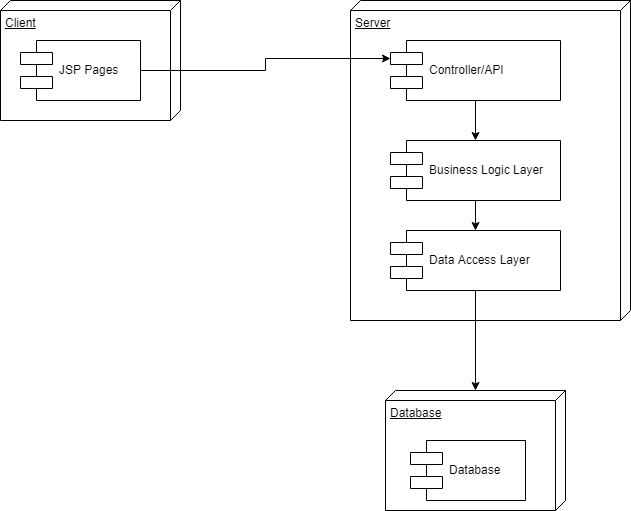
The layers will have the following relationship:



## 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

# Future improvements

# Bibliography