Sport Event Application “Fun&Health”

Analysis and Design Document

Student:Casap Dumitru

**Group:30239**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

*Sport event application “Health&Fun” is a web application which will run on a server. The app is proposed to all people who like sport. The application has two types of users ( a regular user represented by the clients who want to do sport and an administrator user) which have to provide a username and a password in order to use the application.*

*The regular user can perform the following operations:*

* *Create a sport event*
* *View all/near sport events*
* *Update/delete his sport events.*
* *Edit his account.*
* *Select favorite sports.*
* *Create a sport event.*
* *Going/Ungoing to a sport event.*
* *Add comment to a sport event.*
* *Add to friend/ chat with other users*

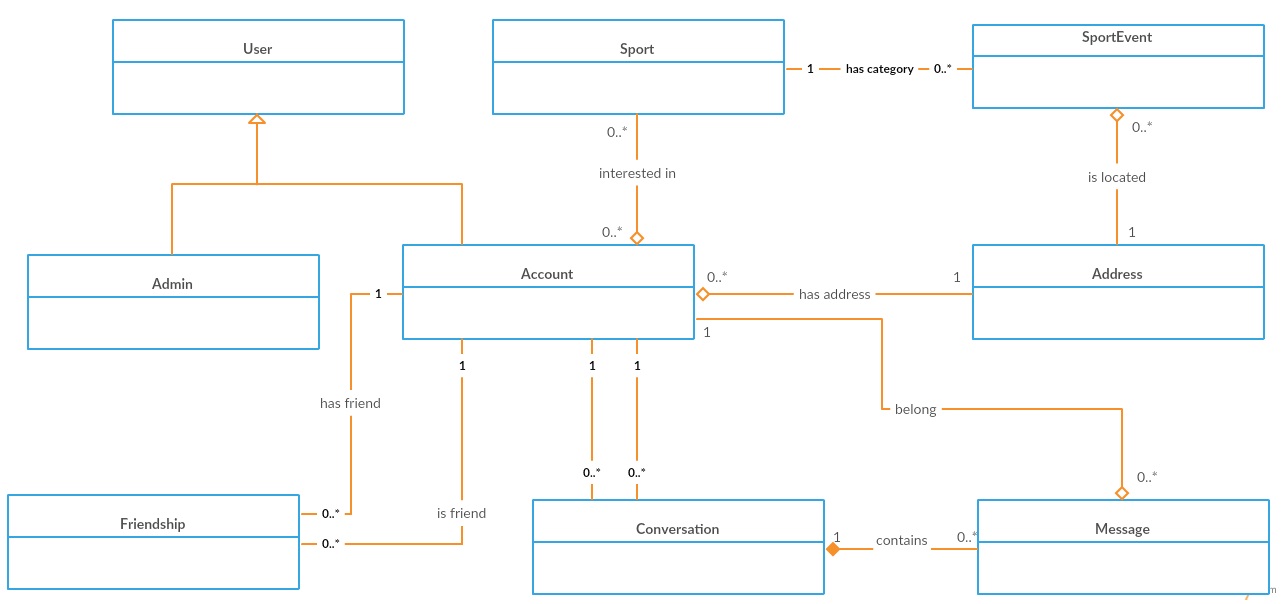
*The administrator can perform the following operations:*

* *Delete sport events*
* *Delete accounts*
* *CRUD on type of sport*
* *Add sport news*

# Elaboration – Iteration 1.1

# Domain Model

A domain model is a system of abstractions that describes selected aspects of a sphere of knowledge. It is a representation of meaningful real-world concepts pertinent to the application that need to be modeled in software. In the case of this application, the domain model are represented by model classes : User, Admin, Account, Sport, SportEvent, Friendship, Conversation, Message, Address and relations between them. The main actor is account, which all be the client of this application. All models are related with Account model. The domain model of the sport event application “Health&Fun” are better explained by the conceptual class diagram.



# Architectural Design

## Conceptual Architecture

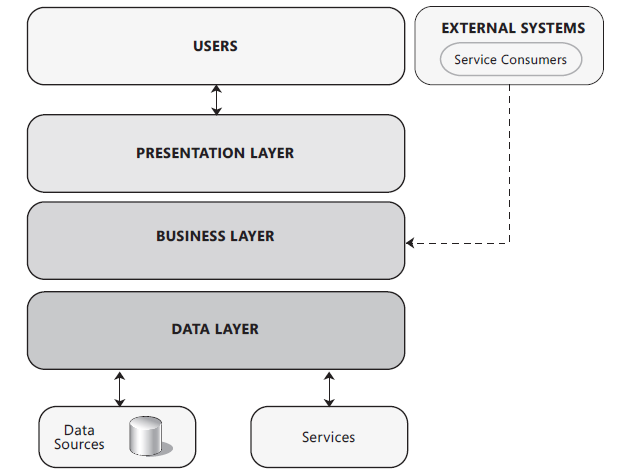
The architectural design pattern used in this project is the layers architectural pattern (Figure 3). An application can consist of a number of basic layers and the common three-layer design consists of the following layers:

**Presentation layer**–  is responsible for the delivery and formatting of information to the application layer for further processing or display. The presentation layer is the lowest layer at which application programmers consider [data structure](https://en.wikipedia.org/wiki/Data_structure) and presentation

**Business layer**– implements the core functionality of the system and encapsulates the relevant business logic

**Data layer** – provides access to data hosted within the boundaries of the system

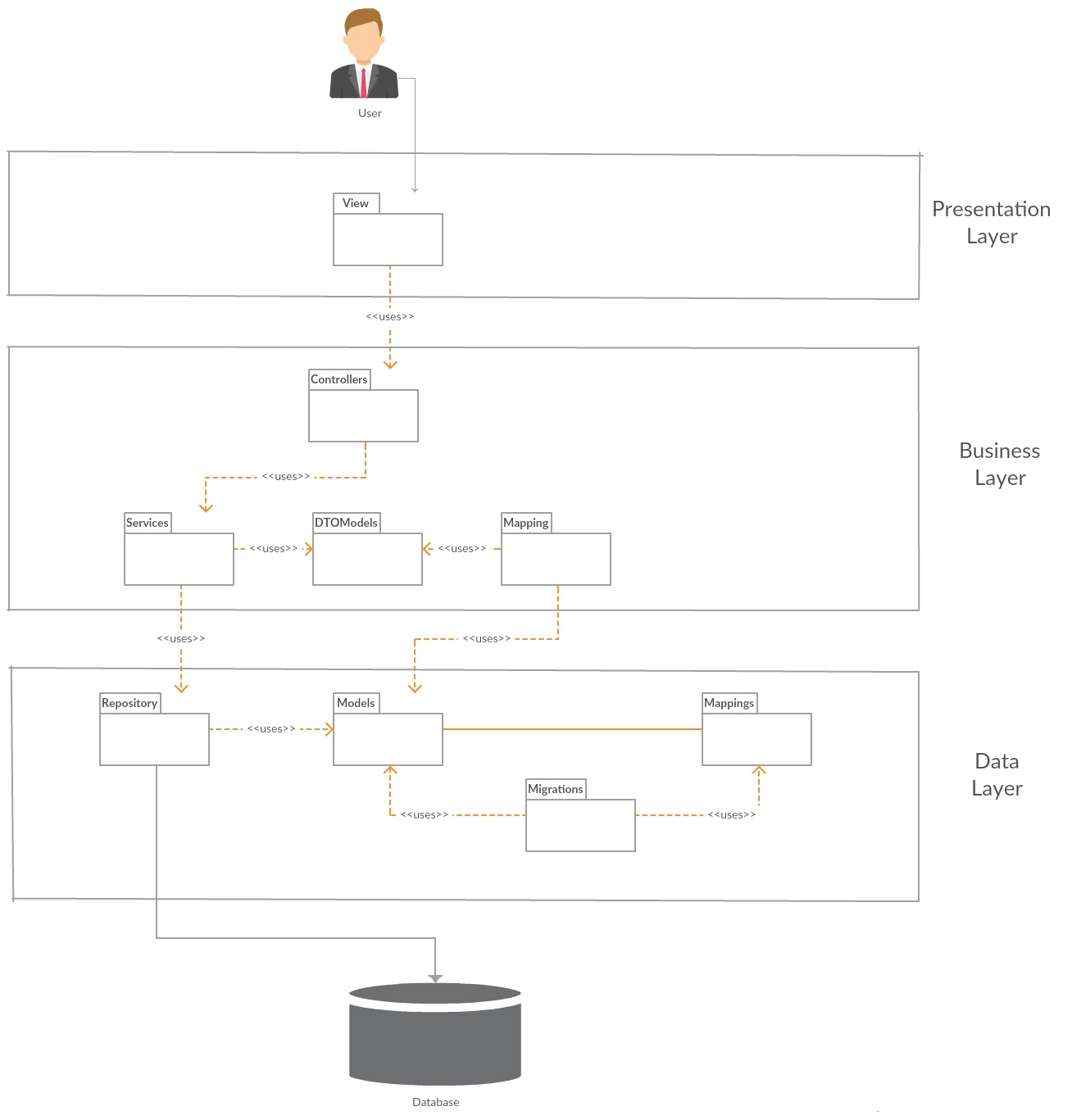
Components within the layered architecture pattern are organized into horizontal layers, each layer performing a specific role within the application. Each layer in the architecture forms an abstraction around the work that needs to be done to satisfy a particular business request. For example, the presentation layer doesn’t need to know or worry about how to get customer data; it only needs to display that information on a screen in particular format. Similarly, the business layer doesn’t need to be concerned about how to format customer data for display on a screen or even where the customer data is coming from; it only needs to get the data from the persistence layer, perform business logic against the data ,and pass that information up to the presentation layer.

**

The layered architecture pattern is a solid general-purpose pattern, making it a good starting point for most applications, particularly when you are not sure what architecture pattern is best suited for your application.

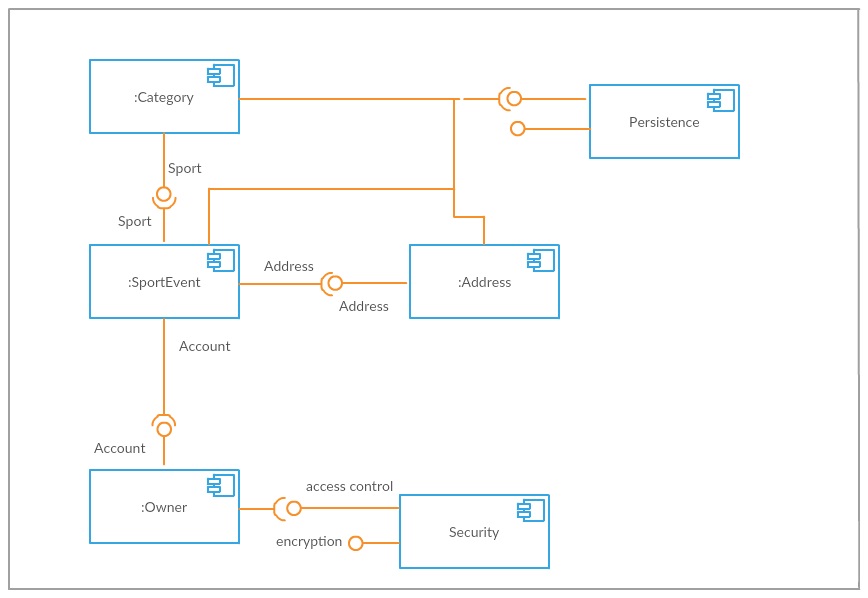
One of the powerful features of the layered architecture pattern is the separation of concerns among components.

## Package Design

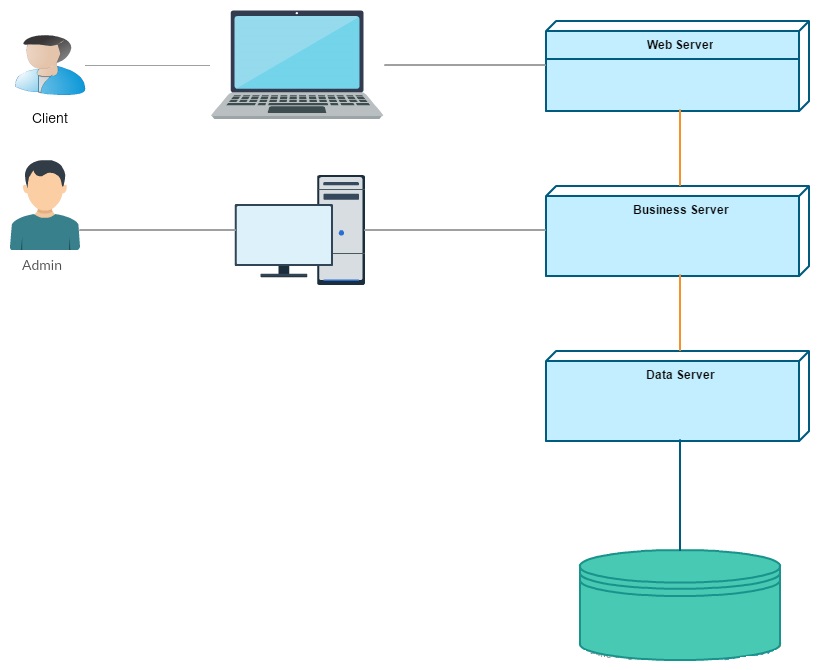


## Component and Deployment Diagrams

Component Diagram



Deployment 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.]*

# 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