

---

**Sportify**

---

**Sportify**  
**Software Architecture Document**

**Version 1.0**

Sportify	Version: 1.0
Software Architecture Document	Date: 20/10/2022>
Sportify sad	

## Revision History

Date	Version	Description	Author
<23/10/22>	<1.0>	First draft	Firangiz Ganbarli

Sportify	Version: 1.0
Software Architecture Document	Date: 20/10/2022>
Sportify sad	

# Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	References	<b>Error! Bookmark not defined.</b>
2.	Architectural Representation	4
3.	Architectural Goals and Constraints	4
4.	Use-Case View	4
4.1	Use-Case Realizations	4
5.	Logical View	4
5.1	Overview	4
5.2	Architecturally Significant Design Packages	4
5.2.1	Design Model: Design Package Diagrams	4
5.2.2	Design Model: Design Package Descriptions	5
5.2.3	Design Model: Design Class Diagrams	5
5.2.4	Design Model: Design Class Descriptions	6
6.	Interface Description	7
7.	Size and Performance	7
8.	Quality	7

Sportify	Version: 1.0
Software Architecture Document	Date: 20/10/2022>
Sportify sad	

# Software Architecture Document

## 1. Introduction

### 1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. Is it intended to capture and convey the significant architectural decisions which have been made on the system.

### 1.2 Scope

This Software Architecture Document provides an architectural overview of Sportify. Sportify will allow users to register and create sport events and will also allow users to search for events that are around them and show the results of that query.

### 1.3 References

1. Sportify – Use Case Specification
2. Sportify – Supplementary Specification
3. Sportify – Use Case Realization

## 2. Architectural Representation

This document presents the architectural as a series of views: use case view, process view, deployment view, and implementation view. These views are presented as Rational Rose Models and use the Unified Modeling Language (UML).

## 3. Architectural Goals and Constraints

Sportify is to be developed as a stand-alone web app that will be accessible by any major Internet Browsers. It consists of four key components: a User Client Module, a Server Module, a Database, and an Administrator Client Module. All components must be able to execute on a personal computer. The User Client and Administrator Client modules must communicate with the server over a TCP/IP connection. The Server and Database components should be located on the same host.

## 4. Use-Case View

The Use Case View is crucial input to the selection of the set of scenarios and use cases that are the focus of an iteration. It describes the set of scenarios and use cases that represent some significant, central functionality. It also describes the set of scenarios and use cases that have many architectural elements.

For more information, refer to Use-Case Specification Requirements document.

### 4.1 Use-Case Realizations

Refer to Use Case Realization document.

## 5. Logical View

### 5.1 Overview

This subsection describes the overall decomposition of the design model in terms of its package hierarchy and layers.

### 5.2 Architecturally Significant Design Packages

#### 5.2.1 Design Model: Packages Diagram

The design model represents the structure and organizations of Sportify. Packages and classes are presented with a brief description.

Sportify	Version: 1.0
Software Architecture Document	Date: 20/10/2022>
Sportify sad	

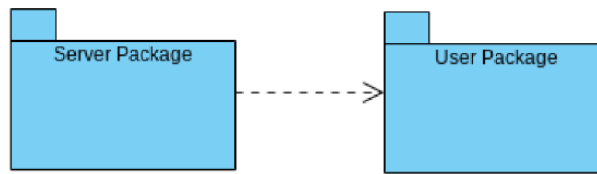


Figure 1: Design Model Package

### 5.2.2 Design Model: Packages Description

Server Package	
Description	This is the main system package where all client queries are managed
Corresponding Classes	Observable, Observer, SportifyServer, DeveloperObserver, AdministratorObserver
Relations	Main package, Dependent of: User
Sub-Packages	Users

User Package	
Description	All info regarding users are handled in this package
Corresponding Classes	User, Event, EventDatabase, SignedUpEvent, SearchResults
Relations	Sub package of Server
Sub-Packages	None

### 5.2.3 Design Model: Design Class Diagram

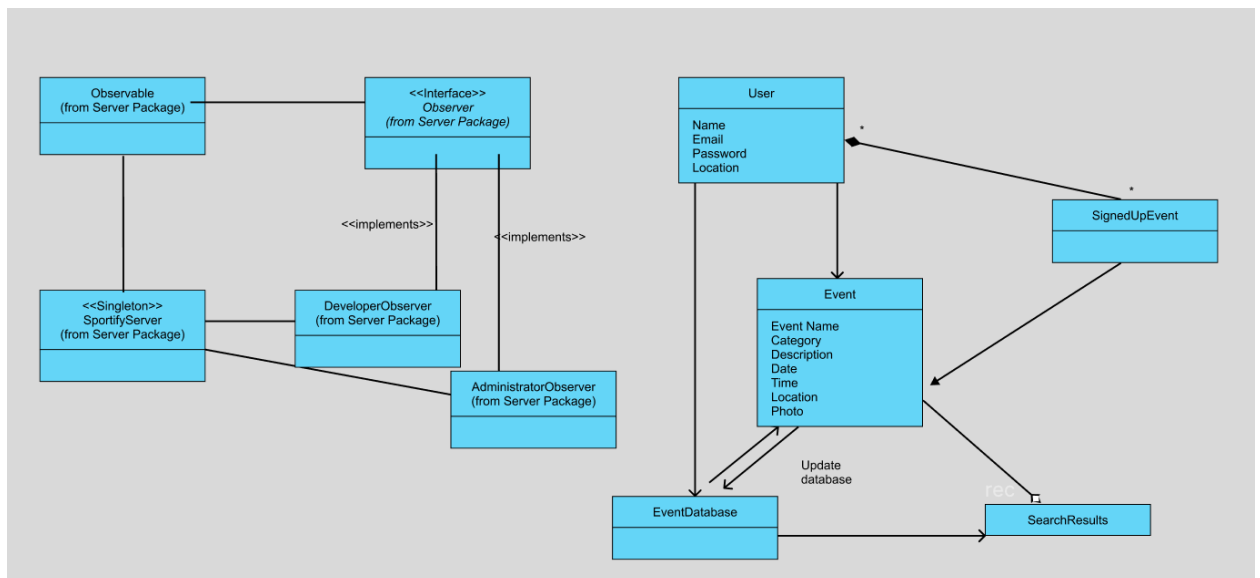


Figure 2: Design Class Diagram

### 5.2.4 Design Classes Description

Sportify	Version: 1.0
Software Architecture Document	Date: 20/10/2022>
Sportify sad	

Property	Description
Name	Observable
Description	Abstract class representing the template observable
Responsibilities	Communicates with SportifyServer and Observer classes

Property	Description
Name	SportifyServer
Description	It's a class that represents the SportifyServer, controls most other classes
Responsibilities	Plays a crucial role in communicating with other classes, and making sure the data passes correctly.

Property	Description
Name	Observer
Description	Abstract class representing the template observer
Responsibilities	Nothing detailed.

Property	Description
Name	DeveloperObserver
Description	Developer implemenation
Responsibilities	Allows the developer to interact with the system easily

Property	Description
Name	AdministratorObserver
Description	Administrator implementation
Responsibilities	Allows the admins of the webpage to control the app as necessary

Property	Description
Name	User
Description	Singleton containing Users lists
Responsibilities	Maintains an updated list of all users

Property	Description
Name	Event
Description	Allows users to create and modify event information
Responsibilities	Main feature of the app, communicates with the EventDatabase to update all the information.

Property	Description
Name	EventDatabase
Description	Singleton containing Events lists
Responsibilities	Maintains an updated list of all events

Property	Description
Name	SearchResults
Description	Displays the search results when a query is received
Responsibilities	Communicates with EventsDatabase class to match the

Sportify	Version: 1.0
Software Architecture Document	Date: 20/10/2022>
Sportify sad	

	query to a list of events to display
--	--------------------------------------

Property	Description
Name	SignedUpEvents
Description	Shows the events that the user has signed up for
Responsibilities	Matches the events to the user and if they are registered, it displays it in the signed up events page.

## 6. Interface Description

Refer to the files within Sportify -> Prototype folder.

## 7. Size and Performance

The selected architecture supports the sizing and timing requirements. The website components have been designed to ensure that it takes minimal time to load and does not slow down the browser significantly.

## 8. Quality

The software architecture supports the quality requirements, as stipulated in the Software Requirements Specifications and Supplementary Specification.