Analysis and Design Document

Student: Bogdan Rogoz
Group: 30433

Revision History

Date	Version	Description	Author
01/04/2018	1.0	Initial documentation	Bogdan Rogoz

Table of Contents

I. Project Specification

The purpose of the Watch2Gether project is to provide its users a friendly environment where they could listen to a wide variety of music, in a synchronized manner, while being situated in totally different spaces.

II. Elaboration – Iteration 1.1

1. Domain Model

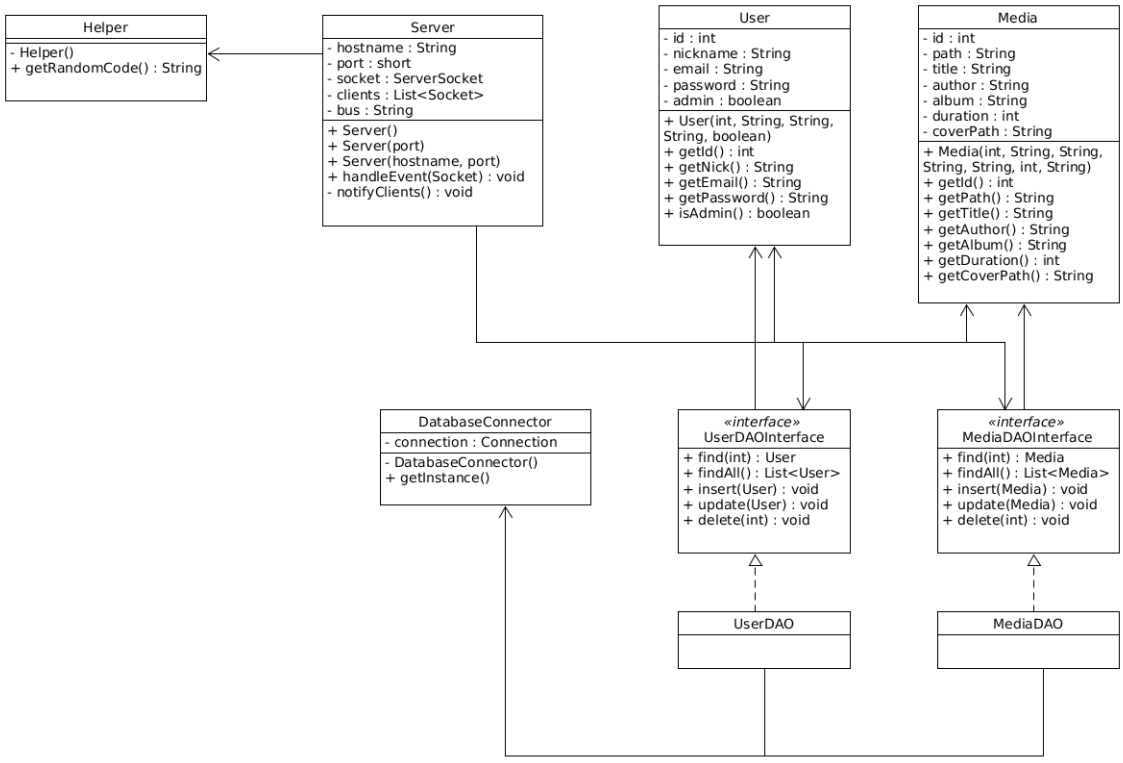
The Client will contain 2 models:

- *User – represents a user, either the active one or other users present in the room*
- *Media – holds the information about media files : title, URL address, cover image, duration etc.*

The Server consists of the following models:

- *User – same as above*
- *Media – same as above*

Although the Client will have a more declarative implementation, the server will have a more object – orietned design, as in the diagram below:



2. Architectural Design

2.1 Conceptual Architecture

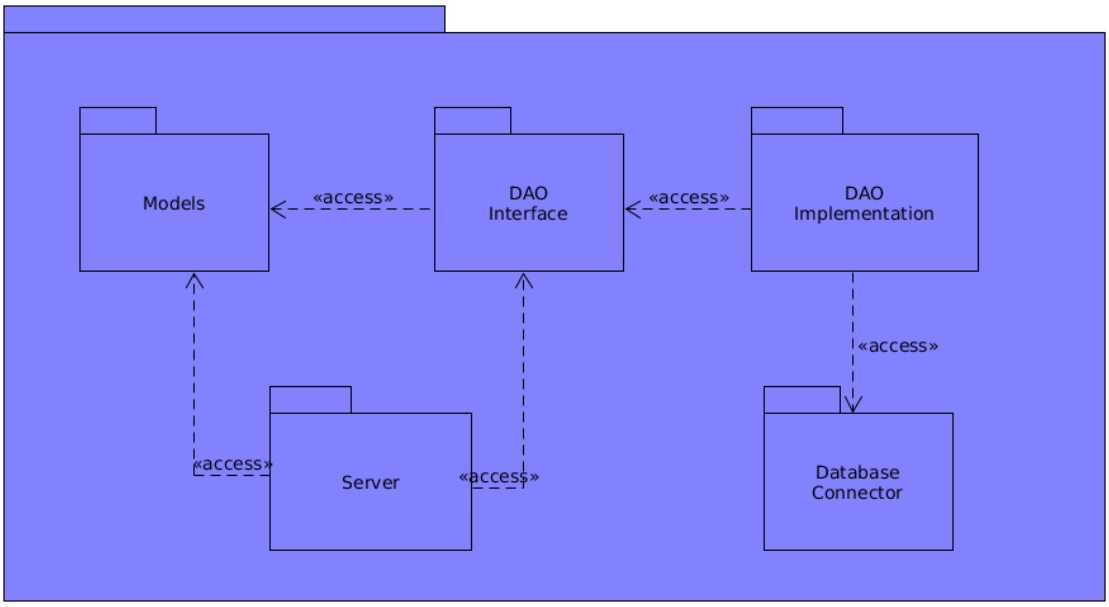
The system consists of two endpoints : the users and the server. The used architectural patterns are:

- Client – Server : The server sends request / response messages to each client at a given point in time and vice-versa. While communicating with all its clients, the server also listens for new connections.

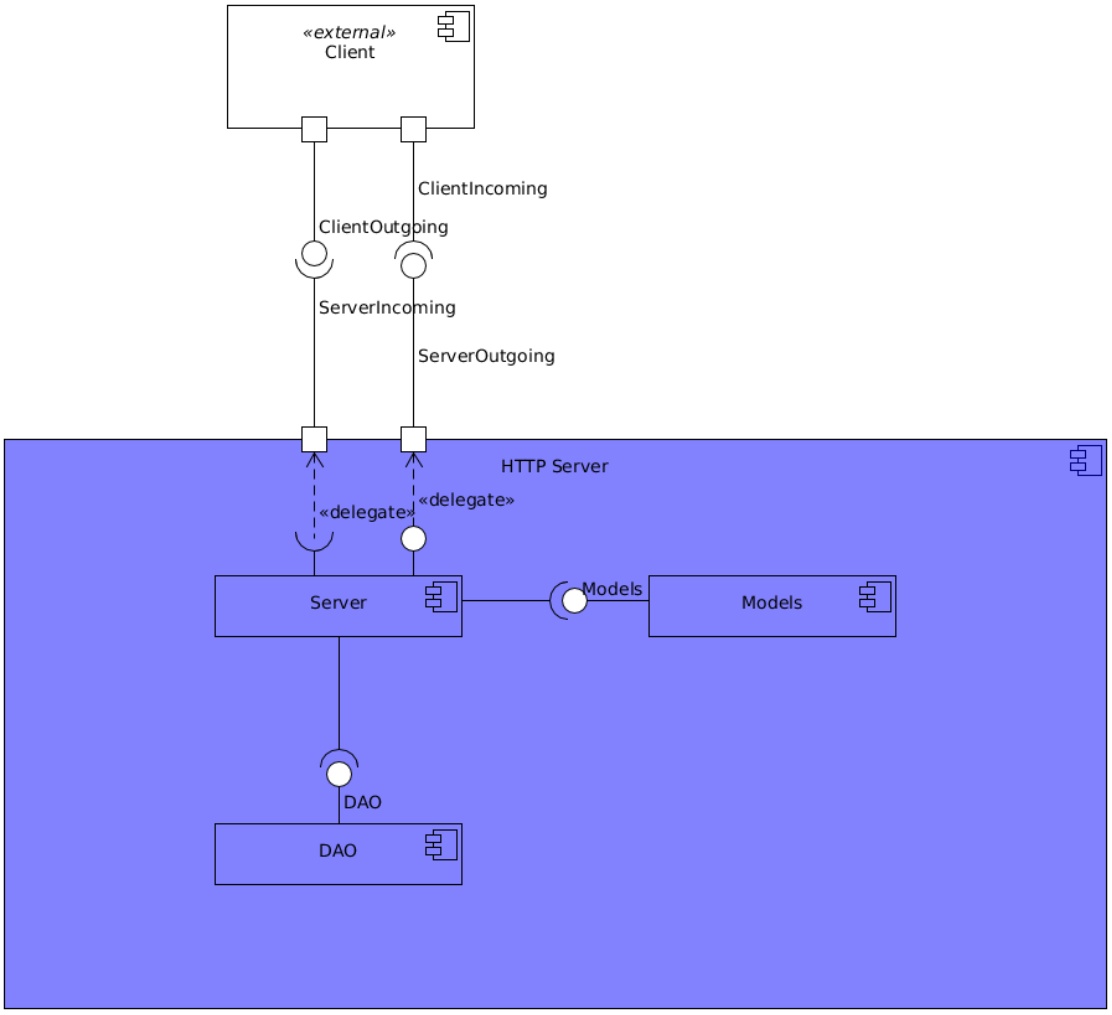
- Event Bus : The server acts as a virtual bus. Every client listens for events while performing playback. When a client wants to perform an action (eg. pause, stop), it sends a message to the server telling it to place the message on the bus, and all the other clients are then notified.

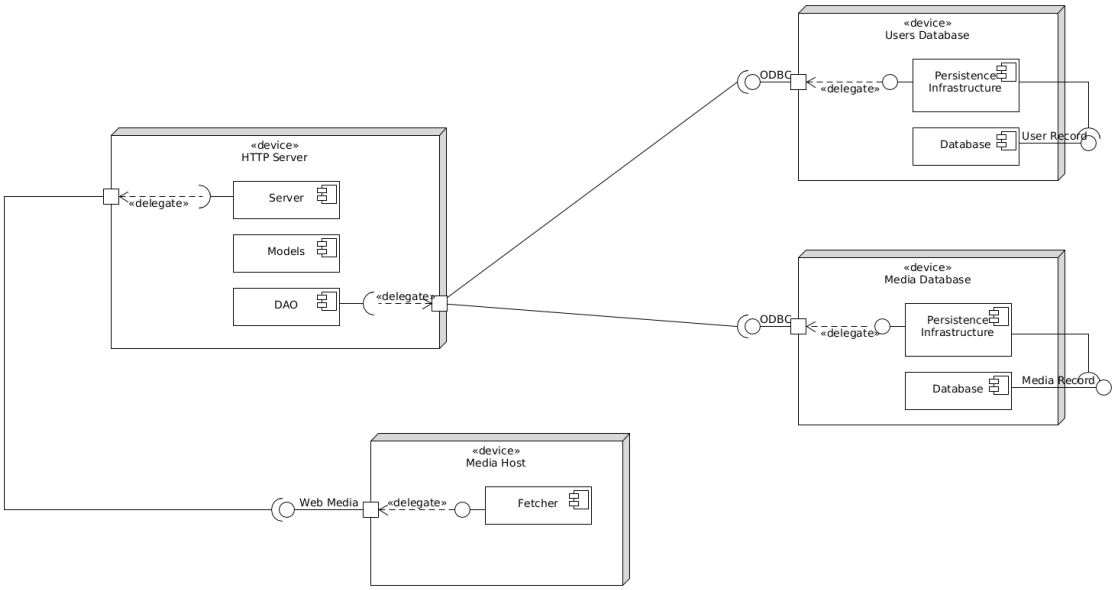
The described approach has been selected due to its simplicity and performance.

2.2 Package Design



2.3 Component and Deployment Diagrams





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